

UNIVERSIDAD DE ALCALÁ

Escuela Politécnica Superior

PROYECTO FIN DE GRADO

**SISTEMA DE MONITORIZACIÓN Y CONTROL DE FINCA
RÚSTICA BASADO EN TECNOLOGÍA WSN**

UNIVERSIDAD DE ALCALÁ

Escuela Politécnica Superior

GRADO EN INGENIERÍA ELECTRÓNICA DE COMUNICACIONES

Trabajo Fin de Grado

**SISTEMA DE MONITORIZACIÓN Y CONTROL DE FINCA
RÚSTICA BASADO EN TECNOLOGÍA WSN**

Autor: Francisco José Espada Moreno

Director: José Manuel Villadangos Carrizo

TRIBUNAL:

Presidente: José Manuel Rodríguez Ascariz

Vocal 1º: Alejandro Martínez Arribas

Vocal 2º: José Manuel Villadangos Carrizo

CALIFICACIÓN:.....

FECHA:.....

AGRADECIMIENTOS

*A mi familia,
por el apoyo y la paciencia que llevan
demostrando durante tanto tiempo*

LISTADO DE ACRÓNIMOS POR ORDEN ALFABÉTICO

ADC: *Analog to Digital Converter*
API: *Application Programming Interface*
BOM: *Bill Of Materials*
BPS: *Bits Per Second*
CSV: *Comma Separated Value*
DMA: *Direct Memory Access*
DNS: *Domain Name System*
FW: *FirmWare*
GPDAIO: *General Purpose Digital-Analog Input-Output*
HTML: *HyperText Markup Language*
HTTP: *Hypertext Transfer Protocol*
HTTPS: *Hypertext Transfer Protocol Secure*
HW: *HardWare*
IoT: *Internet Of Things*
IP: *Internet Protocol*
ISM: *Industrial, Scientific and Medical*
IVA: *Impuesto sobre el Valor Añadido*
LAN: *Local Area Network*
MAC: *Media Access Control*
OTG: *On The Go*
PCB: *Printed Circuit Board*
PVP: *Precio de Venta al Público*
PWM: *Pulse-Width Modulation*
RAM: *Random Access Memory*
SMTP: *Simple Mail Transfer Protocol*
SW: *SoftWare*
TCP: *Transmission Control Protocol*
UART: *Universal Asynchronous Receiver Transceiver*
USB: *Universal Serial Bus*
WPAN: *Wireless Personal Area Network*
WSN: *Wireless Sensor Network*

ÍNDICE DE CONTENIDO

I. RESUMEN	11
I.1. Resumen	12
I.2. Abstract	13
I.3. Estado del arte	15
I.4. Resumen Extendido	18
II. MEMORIA	21
II.1. Tecnología ZigBee	22
II.2. Análisis Energético	25
II.3. Descripción del sistema:	27
II.3.1. Unidad Coordinadora	27
a) Hardware:	27
b) Software:	30
II.3.2. Nodos inalámbricos	31
a) Nodos inalámbricos ZigBee	32
b) Nodos inalámbricos propietarios	35
II.3.3. Sensores y Actuadores	35
II.3.4. Terminales de visualización	37
II.4. Funciones del sistema e interfaz de configuración	39
II.4.1. Procesos	39
II.4.2. Transiciones o condiciones de cambio	41
II.4.3. Estado de las salidas	43
II.5. Servidor Web	44
II.5.1. Interfaz de configuración	45
a) System Overview:	46
b) Process	47
c) Process Config	47
d) GPDAIO	48
e) GPDAIO Configuration	49
f) Wireless Nodes	50
g) Wireless Node Configuration	51
h) Visualization Terminals	52
i) Network Configuration	53
j) E-Mail Notification	54
k) Dynamic DNS	55
II.6. Caso Práctico De Aplicación	56
II.6.1. Escenario bajo monitorización y control	56
II.6.2. Nodos inalámbricos del sistema	57
II.6.3. Configuración de las GPDAIOs	58
II.6.4. Procesos que intervienen en el sistema	59
a) Control Depósito	59

b) Riego Césped	60
c) Riego Frutales	61
d) Depuración	62
e) Alarma	64
II.6.5. Terminal de visualización	66
II.7. Resultados y Conclusiones	67
II.8. Trabajo Futuro	68
III. DIAGRAMAS Y PLANOS	71
III.1.Unidad Coordinadora	72
III.1.1.Tarjeta PIC32 Ethernet Starter Kit	72
III.1.2. Tarjeta Starter Kit IO Expansion Board	79
III.1.3.Tarjeta PICtail Daughter Board for SD and MMC Cards	80
III.2. Prototipo de nodo inalámbrico	81
IV. SOFTWARE	83
IV.1. Principales Estructuras de los Objetos de Red	84
IV.2. Gestión de nodos inalámbricos genéricos	86
IV.3. Generación de código HTML dinámico en servidor Web	87
IV.4. Página Web con funciones JavaScript y variables dinámicas	89
V.PRESUPUESTO	95
V.1. Diseño y Desarrollo Hardware y Software	96
V.2. Fabricación de Prototipos	98
V.3. Coste Total: Desarrollo + Prototipos	99
VI. LINKS Y BIBLIOGRAFÍA	101

ÍNDICE DE FIGURAS

Figura 1. Evolución de la integración de la tecnología ZigBee en distintos campos durante los últimos años y previsiones a corto plazo	15
Figura 2. Ejemplo de integración de tecnologías WSN (WPAN) y LAN en aplicaciones médicas	16
Figura 3. Integración de unidades de medida, monitorización y actuación en celdas de distribución eléctrica (proyecto e-DIANA)	16
Figura 4. Ejemplo de sistemas que formarán parte del IoT en el hogar y funciones de cada uno	17
Figura 5. Ejemplo de aplicación del sistema (ver sección “II.6 Caso Práctico De Aplicación”)	18
Figura 6. Esquema general de los componentes del sistema	20
Figura 7. Frecuencias de operación de la tecnología ZigBee, anchos de banda y canales	23
Figura 8. Comparativa de anchos de banda de las distintas tecnologías en bandas ISM más comunes	23
Figura 9. Coste energético para un mensaje de poll con o sin transferencia de datos desde nodo enrutador (medida en mV equivalente a $I_{cc} \times 10\Omega$).	25
Figura 10. Prototipo de unidad coordinadora utilizada durante el desarrollo	27
Figura 11. Tarjeta PIC32 Ethernet Starter Kit	28
Figura 12. Tarjeta de expansión con núcleo procesador y elementos expansores integrados	29
Figura 13. Módulo de comunicaciones ZigBee ETRX3	29
Figura 14. Esquema de bloques interno del módulo radio ETRX3	30
Figura 15. PICtail almacenamiento de información en tarjetas SD	30
Figura 16. Bloques internos de la unidad coordinadora	31
Figura 17. Bloques internos de los nodos inalámbricos ZigBee basados en el módulo ETRX2 y ETRX3	33
Figura 18. Adaptador 4-20mA para sensores de temperatura tipo PT100	35
Figura 19. Sensor de temperatura basado en LM35	36
Figura 20. Sensor de humedad basado en el amplificador operacional LM393 mas hygristor HR31	36
Figura 21. HC-SR501: Sensor de presencia PIR	36
Figura 22. MQ-2: Sensor de gas butano/metano	36
Figura 23. Módulo de receptor láser (ej. receptor para barrera láser)	37
Figura 24. Módulo de actuación mediante relé	37
Figura 25. Ejemplo de tableta básica de bajo coste utilizada como terminal de visualización	38
Figura 26. Objetos a nivel de procesamiento. Los GPDAIOs se utilizan como enlace entre los nodos (dependientes del HW) y los procesos (genéricos)	39
Figura 27. Máquina de estados con transiciones entre procesos y principales condiciones de cambio	40
Figura 28. Procesamiento de los niveles de condición dentro de un estado	41
Figura 29. Ejemplo de rutina de envío de código dinámico HTML en función de los terminales de visualización configurados	45
Figura 30. Páginas Web de configuración y enlaces internos	46
Figura 31. Página de información general del sistema	46
Figura 32. Web con el listado de procesos activos en el sistema	47
Figura 33. Página de configuración de procesos y estados de los mismos	48
Figura 34. Web con el listado de GPDAIOs configuradas en el sistema	49

Figura 35. Página de configuración de parámetros de GPDAIOs	50
Figura 36. Web con listado de nodos inalámbricos configurados en el sistema	51
Figura 37. Web de configuración de parámetros de nodos inalámbricos	52
Figura 38. Web de configuración de datos a representar en los terminales de visualización	53
Figura 39. Web de configuración de los parámetros de red	54
Figura 40. Web de configuración de los parámetros de correo saliente SMTP	55
Figura 41. Configuración de los parámetros para el uso de servicios de DNS dinámicos	55
Figura 42. Escenario real de aplicación del sistema	56
Figura 43. Módulos radio transceptores de 100mW (izquierda) y 1W (derecha) de potencia a 433 Mhz basado en el integrado SI4432 (Silicon Labs) para ser añadido como segundo interfaz radio.	69

ÍNDICE DE TABLAS

Tabla 1. Coste energético (equivalente mA) por tipo de mensaje ($V_{cc} = 3.3v$)	26
Tabla 2. Estimación de duración de batería tipo CR123A en función de los tiempos de transmisión	26
Tabla 3. Compatibilidades y tipos de salida de alimentación en función de la fuente de alimentación de entrada	34
Tabla 4. Estados y funciones de LEDs indicadores	35
Tabla 5. Resumen de tipos de factores permitidos como entradas de un estado, tipo de condición y factores de comparación permitidos.	42
Tabla 6. Factores y condiciones permitidas según el tipo de salida seleccionada	43
Tabla 7. Listado de nodos inalámbricos configurados en el escenario bajo monitorización	57
Tabla 8. Listado de GPDAIOs configurados en el escenario bajo monitorización	58
Tabla 9. Condiciones de entrada y niveles de salida para cada uno de los estados del proceso de control de llenado de depósito	60
Tabla 10. Condiciones de entrada y niveles de salida para cada uno de los estados del proceso de control de riego de césped	61
Tabla 11. Condiciones de entrada y niveles de salida para cada uno de los estados del proceso de control de riego de árboles frutales	62
Tabla 12. Condiciones de entrada y niveles de salida para cada uno de los estados del proceso de control de depuración	63
Tabla 13. Condiciones de entrada y niveles de salida para cada uno de los estados del proceso de control alarma	65
Tabla 14. Ejemplo de terminal de visualización adaptado a la aplicación de control para finca rústica	66

I. RESUMEN

I.1. Resumen

Desde hace aproximadamente una década, la tecnología basada en redes de sensores inalámbricos (WSN) se ha introducido con fuerza en el mercado. Este tipo de dispositivos se han diseñado tanto para uso doméstico como industrial, aportando gran flexibilidad al despliegue de redes para transmisión de datos en sistemas de monitorización y control.

Sin embargo, la transparencia y flexibilidad proporcionadas por esta tecnología en las capa física, de comunicaciones y gestión de red, no se traslada hacia las capas de aplicación e interacción con el usuario. El sistema desarrollado en este TFG permite la integración de dispositivos WSN en prácticamente cualquier tipo de aplicación, permitiendo al usuario adaptar la tecnología directamente a sus necesidades con total flexibilidad y sin la necesidad de conocimientos técnicos.

I.2. Abstract

During the last decade, a significant amount of systems based on Wireless Sensor Networks (WSN) has been introduced into the market. Devices based on this technology has been designed for their integration and use in both domestic and industrial fields, providing huge flexibility and reliability in data transmission and communication network implementation.

However, the flexibility and transparency provided in low level (physical, transport and network management), have not been transferred to application and user interface layers. The system developed along this end-of-degree project attempts to bridge this lack, providing to the user the capability of using the benefits of this technology directly on their own applications with total flexibility and without the need of technical knowledge.

I.3. Palabras clave

Wireless Sensor Network (WSN)

Monitorización y control inalámbrico

ZigBee

Internet Of Things

I.4. Estado del arte

En los últimos años el mercado ha experimentado una gran evolución de los sistemas de automatización y control tanto en el ámbito industrial como doméstico. La reducción de costes y una mejora de prestaciones en la mayoría de los elementos que forman los mismos han permitido su expansión y utilización en aplicaciones cada vez más sencillas sin suponer un incremento significativo en el coste.

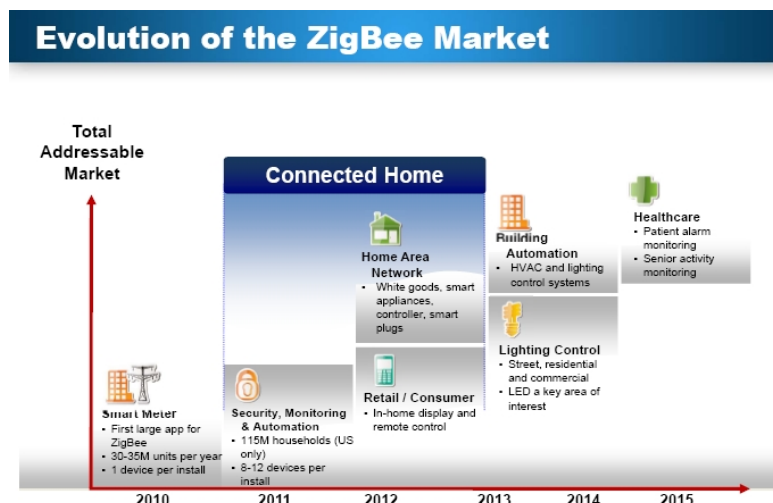


Figura 1. Evolución de la integración de la tecnología ZigBee en distintos campos durante los últimos años y previsiones a corto plazo

La reciente evolución de las tecnologías inalámbricas, ha permitido una mayor integrabilidad y una menor invasión del escenario a monitorizar, a lo cual, el mercado ha reaccionado con una gran demanda de dispositivos de este tipo. Con el fin de compatibilizar sistemas de distintos fabricantes, aparecieron alianzas protocolarias, la mayoría basadas en el estándar IEEE 802.15.4 (lanzado en 2007). La más conocida de denomina ZigBee, de la cual se han lanzado 3 versiones desde su primer borrador el año 2003.

Estas pilas de protocolos permiten la creación de redes de sensores inalámbricos (WSN, *Wireless Sensor Network*), en las cuales conviven multitud de "motas" inalámbricas o nodos, los cuales pueden tener uno o más sensores y/o actuadores asociados. El protocolo gestiona tanto la organización de la red, creación de la misma, gestión de miembros, envío de mensajes punto-punto y punto-multipunto, etc. Además, el uso de la IEEE 802.15.4 junto con un hardware adecuado, permite obtener gran eficiencia desde el punto de vista energético.

Este tipo de tecnología se ha aplicado en diversos campos, tales como la medicina, donde es habitual el uso de redes de comunicaciones aisladas, y que tecnologías como WPAN y WLAN juegan un papel fundamental para la integración de todas las comunicaciones en una red heterogénea, integrada y global, que permita tanto un mejor cuidado del paciente con monitorización en tiempo real de las constantes, centralización de datos, así como un intercambio de información entre la comunidad médica.

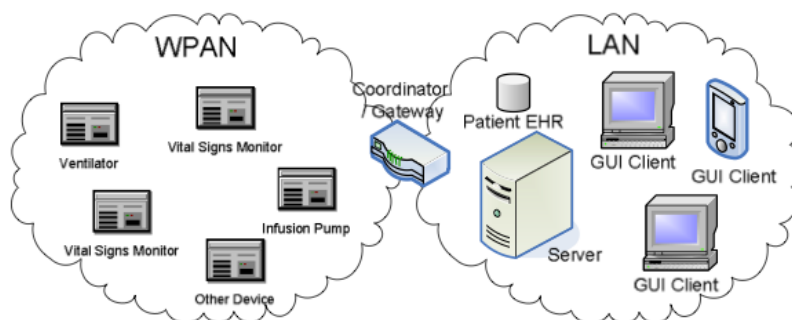


Figura 2. Ejemplo de integración de tecnologías WSN (WPAN) y LAN en aplicaciones médicas

Otro campo de aplicación en el que ZigBee aporta grandes ventajas es en la eficiencia energética de edificios. En Europa, el 40% del total de la energía consumida es demandada por los edificios, siendo superior a la demanda de sectores tan potentes como transporte o industria. Proyectos como eDIANA, “Embedded Systems for Energy Efficient Buildings”, financiados por la unión Europea dentro del marco ARTEMISIA (séptimo programa marco, FP7), han demostrado la apuesta de las instituciones por la integración de redes WSN, incluso, en los servicios domésticos.

Este tipo de proyectos se centran en incrementar la eficiencia del uso de los recursos energéticos, proporcionando un mejor servicio de los mismos, e infraestructuras mas eficientes y fiables.

En este campo, ZigBee proporciona una forma sencilla, fiable, estándar, de bajo coste y no intrusiva para el despliegue de la red de monitorización energética, así como de actuación sobre la misma.

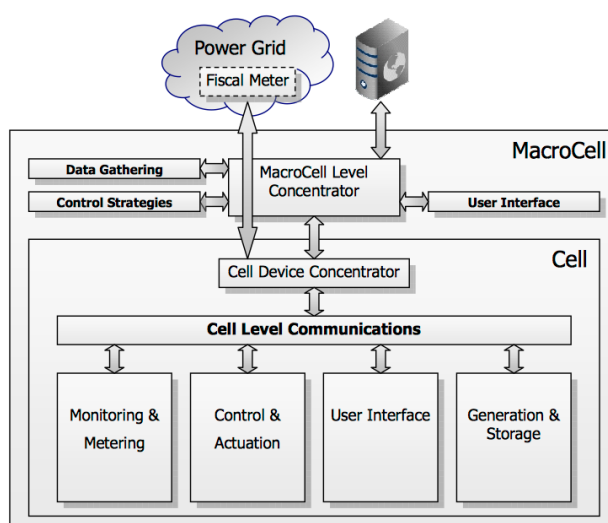


Figura 3. Integración de unidades de medida, monitorización y actuación en celdas de distribución eléctrica (proyecto e-DIANA)

ZigBee y las redes WSN han sido predecesoras, y formarán parte del siguiente paso en la integración de redes, la llamada “Internet of Things” (IoT). Este concepto es un paso mas en el nivel de abstracción de las WPAN, en el que las redes pasan de conectar “personas” (*personnel*) a conectar cosas (*things*). Con la reducción de costes en el HW de los módulos, la estandarización y la distribución mundial de los mismos, ZigBee forma parte del grupo de tecnologías que hacen posible el IoT, proporcionando conectividad e interconectividad “ilimitada” a prácticamente cualquier “cosa”. Ericsson prevee 50.000M de dispositivos conectados a Internet para 2020, lo que equivale a 10 unidades por persona, solo contando aquellas personas que disponen de teléfono móvil (personas activas tecnológicamente). Aunque el *gateway* a Internet debe ser proporcionado por otra tecnología

(ejemplo, WLAN), ZigBee resuelve la base de la estructura piramidal del conjunto de protocolos del sistema.

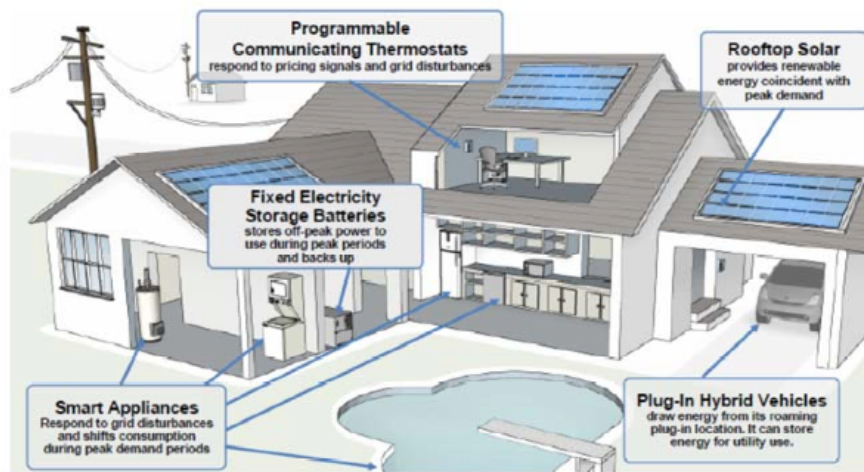


Figura 4. Ejemplo de sistemas que formarán parte del IoT en el hogar y funciones de cada uno

Sin embargo, la utilización de este tipo de tecnologías requiere, en la mayoría de los casos, de un integrador/desarrollador intermedio que diseñe y particularice el sistema a cada escenario según los requisitos específicos de cada aplicación. Aunque alianzas como ZigBee y pilas como la IEEE 802.15.4 proporcionan transparencia a bajo nivel, existe la carencia de un sistema que facilite la adaptabilidad de la tecnología a nivel de usuario (alto nivel) sin necesidad de la intervención de terceros ni conocimientos técnicos.

Este TFG pretende dar soporte a esta necesidad, adaptando las tecnologías basadas en WSN al usuario final, de forma que el mismo pueda moldear el sistema a cada aplicación en particular de una forma totalmente intuitiva y sencilla, y a la vez, conservando la flexibilidad que este tipo de tecnologías proporciona.

I.5. Resumen Extendido

El trabajo a realizar durante este TFG consiste en el desarrollo de un sistema de gestión y configuración de redes WSN que permite directamente al usuario final adaptar y usar este tipo de tecnologías, en constante desarrollo, a una aplicación en particular sin necesidad de la intervención de terceras partes, tales como desarrolladores intermedios.

Dotando al usuario final de la capacidad de adaptar el sistema a sus necesidades, se proporciona incluso a alto nivel la máxima flexibilidad que permite este tipo de tecnologías a la vez que disminuye el conocimiento técnico necesario para su utilización. Además contribuye a una disminución significativa de costes, que a su vez que promueve la expandibilidad de la tecnología en el mercado y posibilita la apertura de puertas a nuevas áreas y campos de aplicación.

El sistema planteado se desarrollará tanto desde el punto de vista software como hardware. Si bien, con el fin de ahorrar costes en el desarrollo, parte del hardware utilizado estará basado en tarjetas de desarrollo comerciales, las cuales integran el núcleo de procesamiento de la unidad principal (coordinadora) y algunos periféricos como puertos Ethernet o USB. Además, puesto que se trata de tarjetas de desarrollo proporcionadas por los fabricantes de circuitos integrados, la adaptación del diseño y la fabricación del mismo para un futuro producto comercial, y propietario, resulta muy sencilla, ya que toda la información está disponible (ej. esquemas eléctricos). A su vez, esta metodología de trabajo, disminuye el *time to market*, lo cual potencia los beneficios a corto y medio plazo. No obstante, otros elementos hardware como los prototipos de nodos inalámbricos han sido desarrollados durante este TFG.

Desde el punto de vista software, todo el sistema estará programado utilizando una filosofía propietaria que optimiza el uso de recursos y adapta al máximo el código al tipo de hardware utilizado. El desarrollo de un software a medida a bajo nivel (C e incluso ensamblador), permite al sistema ejecutar sus funciones sobre una plataforma hardware de reducidos recursos gracias a su gran eficiencia, ahorrando costes y potenciando aun más una posible salida comercial del producto. No hay que olvidar, que el objetivo es la utilización masiva del sistema en todo tipo de aplicaciones, incluso domésticas, para lo cual es necesario que el coste sea lo más reducido posible.

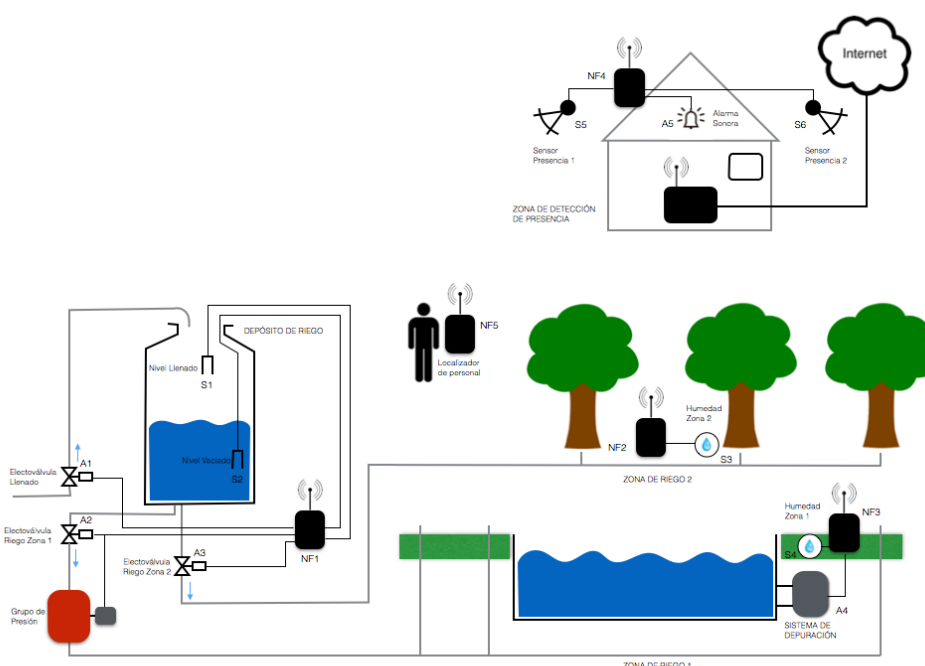
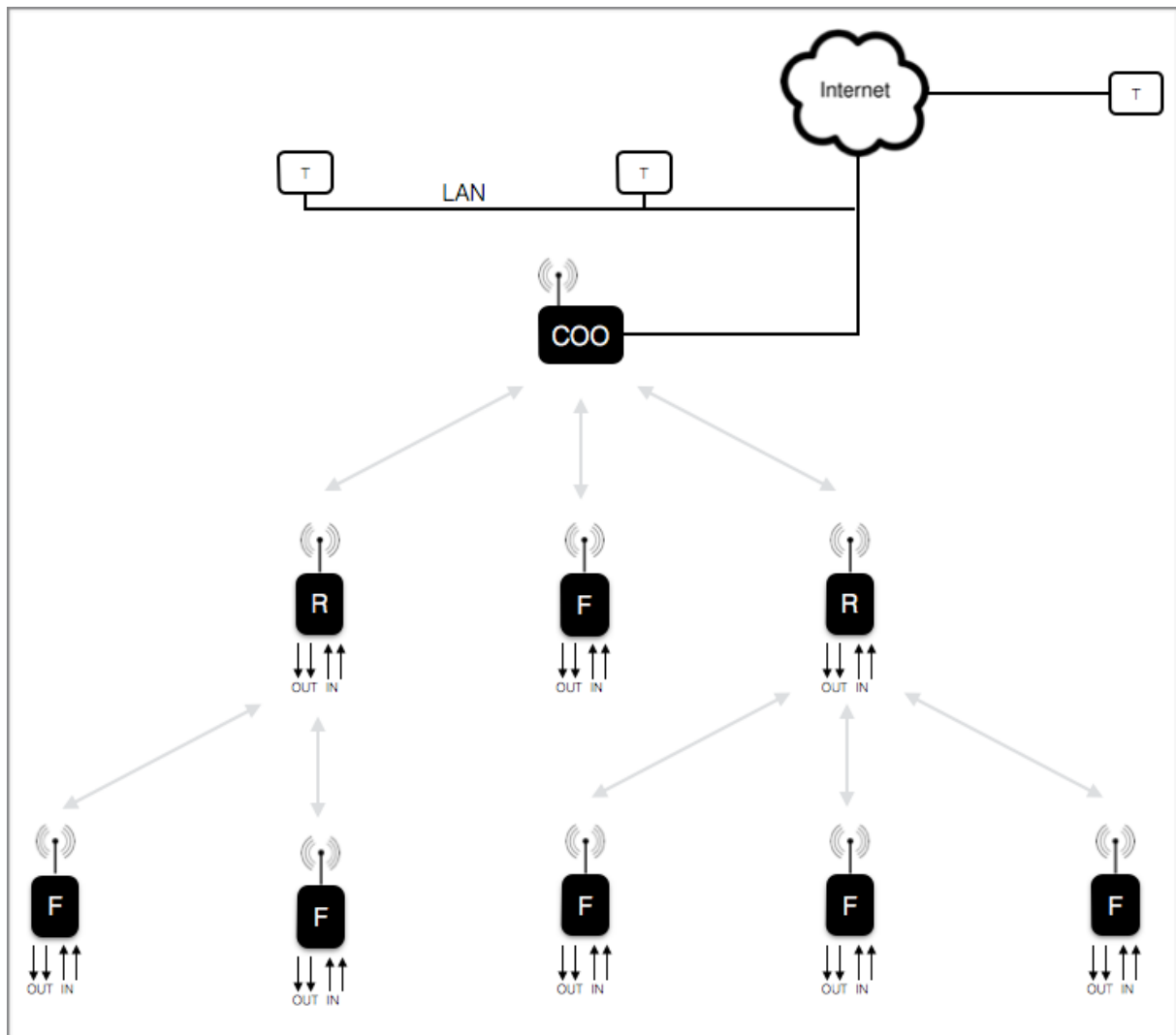


Figura 5. Ejemplo de aplicación del sistema (ver sección “II.6 Caso Práctico De Aplicación”)

El sistema se divide en las siguientes partes:

- Unidad coordinadora: Se trata del núcleo de procesamiento principal, encargado de tomar decisiones en función de la configuración establecida y recibir las peticiones procedentes del usuario. A su vez, incorpora periféricos, como un puerto Ethernet, que le permite la conexión a redes LAN y posibilita el acceso remoto, tanto a los datos como a la configuración, a través de Internet. Este dispositivo necesita alimentación continua, y no dispone de modos de bajo consumo, por lo que deberá estar conectado a la red de 230VAC.
- Nodos inalámbricos: Distribuidos a lo largo del escenario bajo monitorización y/o control, proporcionan datos al coordinador sobre los sensores y actuadores conectados a los mismos, a la vez que reciben información desde este. Son elementos de bajo coste, pequeño tamaño y ultra bajo consumo. En una misma aplicación pueden existir multitud de estos elementos (el número dependerá de la complejidad de la misma). La mayoría de ellos pueden ir alimentados por baterías con una autonomía de meses o años. Aquellas unidades que requieran mayor cantidad de energía, deberán ir conectadas a la red de 230VAC o baterías recargables o de gran capacidad.
- Sensores y actuadores: estos componentes del sistema estarán conectados a los nodos inalámbricos de forma que proporcionen información sobre el escenario bajo monitorización y permitan al sistema actuar sobre elementos externos. Sensores de tipo humedad, temperatura, presencia, etc. son habituales en gran cantidad de aplicaciones y deberá ser el usuario quien los conecte al sistema en función de las necesidades particulares de cada caso. Algunos actuadores simples como relés de baja potencia o elementos similares pueden ir integrados en los propios nodos inalámbricos.
- Terminales de visualización: Se encargan de proporcionar datos al usuario sobre el estado del sistema tales como lecturas de sensores, estado de los actuadores, decisiones lógicas tomadas, etc. Estos dispositivos son *tablets* o móviles comerciales, ya que todo el entorno de visualización está basado en Web. Pueden existir tantos terminales de visualización como se requieran, situados en el entorno de la unidad coordinadora o remotamente con conexión a través de Internet.



COO: Unidad coordinadora
 R: Nodo inalámbrico enrutador
 F: Nodo inalámbrico final
 T: Terminal de visualización

Figura 6. Esquema general de los componentes del sistema

II. MEMORIA

II.1. Tecnología ZigBee

ZigBee es un protocolo de comunicaciones inalámbricas basado en el estándar IEEE 802.15.4, y que se presenta como una alternativa frente a otros protocolos ya existentes. El origen de ZigBee lo encontramos en la antigua alianza HomeRF, un consorcio de compañías de comunicación wireless, disuelta en 2003, que creó el estándar HomeRF. El objetivo de dicho estándar era crear un protocolo para el intercambio de datos entre dispositivos distribuidos en una vivienda, idea que recogió la tecnología ZigBee.

A finales de 1998 se empezaron a concebir las primeras ideas sobre ZigBee. El objetivo era satisfacer las necesidades existentes en una serie de aplicaciones en las que las tecnologías ya existentes no presentaban una solución viable. A partir de aquí, nació, en 2002, la ZigBee Alliance, una asociación sin ánimo de lucro de más de 175 empresas. El objetivo de este consorcio es proveer a los usuarios con los últimos avances en cuanto a flexibilidad, movilidad y uso de redes inalámbricas e introducir la tecnología ZigBee en una amplia gama de productos y aplicaciones.

Este grupo de empresas que ha desarrollado ZigBee, formó el ZigBee Working Group, en el que también participó el grupo de trabajo 4 del comité de estandarización IEEE 802.15. Este grupo es el encargado del desarrollo de especificaciones para productos PAN (*Personal Area Network*) en las bandas de frecuencia de uso sin licencia. De esta manera, el estándar IEEE 802.15.4 quedó completado en 2003 y fue ratificado a finales de 2004, presentando la versión 1.0 de ZigBee en Junio de 2005, e incluyendo algunas mejoras de seguridad en 2006.

La última versión, denominada ZigBee PRO, y sobre la que trabaja el sistema desarrollado en este TFG, fue lanzada en 2007.

Las características del protocolo que más cabría destacar son su flexibilidad, su bajo coste y bajo consumo de energía, que a su vez son las características que conforman la idea básica de la que nació ZigBee: dispositivos fáciles de manejar que permiten formar redes inalámbricas de bajo coste, y que, gracias al bajo consumo, permiten una amplia variedad de aplicaciones.

En general, se puede decir que una de las principales ventajas que presenta el protocolo ZigBee frente a otros, es que reduce el tiempo en el que la comunicación vía radio está activa, y, por tanto, reduce considerablemente el consumo. En el caso de un nodo final (tipo "F"), únicamente debe permanecer activo en el instante en el que se produce el envío o recepción de datos desde las balizas o routers (tipo "R"). Desde este punto de vista, se hablará de un consumo asimétrico, ya que algunos dispositivos, como en el caso de los nodos finales, tendrán un consumo muy reducido, mientras que otros, como los coordinadores de la red y routers, deben permanecer siempre activos y, por tanto, consumirán notablemente más energía que los anteriores.

El estándar especifica que ZigBee puede trabajar en tres bandas ISM: 868 MHz, 915 MHz y 2,4 GHz. La mayoría de los fabricantes optarán por la última, ya que puede ser usada en todo el mundo, mientras que las dos primeras sólo se pueden usar en Europa y EEUU, respectivamente y sus prestaciones son inferiores en cuanto a ancho de banda se refiere, y consecuentemente a consumos de dispositivos finales (tipo "F").

En el caso de la banda de 868 MHz la capa física soporta un único canal comprendido entre los 868 y 868.6 MHz, mientras que, para los 915 MHz tendríamos diez canales (situados entre los 902 y 928 MHz). En total tenemos los 11 primeros canales que nos permite utilizar ZigBee.

Para 2,4 GHz, la más utilizada, tenemos 16 canales (con 5 MHz de espacio entre canales) comprendidos entre 2,4 y 2,4835 GHz.

Frequency Band	License Required?	Geographic Region	Data Rate	Channel Number(s)
868.3 MHz	No	Europe	20kbps	0
902-928 MHz	No	Americas	40kbps	1-10
2405-2480 MHz	No	Worldwide	250kbps	11-26

Figura 7. Frecuencias de operación de la tecnología ZigBee, anchos de banda y canales

En la figura anterior también podemos observar las tasas de transferencia correspondientes a cada banda de frecuencia. El hecho de que estas tasas sean menores en comparación con las de otros protocolos de comunicación wireless simplifica el procesamiento de la señal, y, en consecuencia, reduce el coste de los módulos.

Existen muchas otras tecnologías hoy en día que nos permiten formar redes inalámbricas. Entre ellas cabría destacar las que comparten la banda de 2,4 GHz con ZigBee: Bluetooth y WiFi, así como protocolos propietarios tipo MiWi.

Desde un primer momento se trató al estándar ZigBee como una tecnología WPAN (*Wireless Personnel Area Network*) debido a su menor alcance en comparación con Bluetooth o WiFi.

Bluetooth y ZigBee tienen mucho en común, ambos son protocolos de redes de alcance personal. Pero Bluetooth tiene una tasa de transmisión muy superior (1 Mbps) ya que está más orientado a eliminar la necesidad de cables para comunicarse entre dispositivos electrónicos y accesorios, como puede ser el caso de un ordenador y unos auriculares inalámbricos, y además permite el intercambio de archivos entre ellos.

Otra diferencia significativa es el consumo. Los dispositivos Bluetooth consumen más que los módulos ZigBee, ya que los primeros deben enviar información de forma continua a la red para mantener el sincronismo.

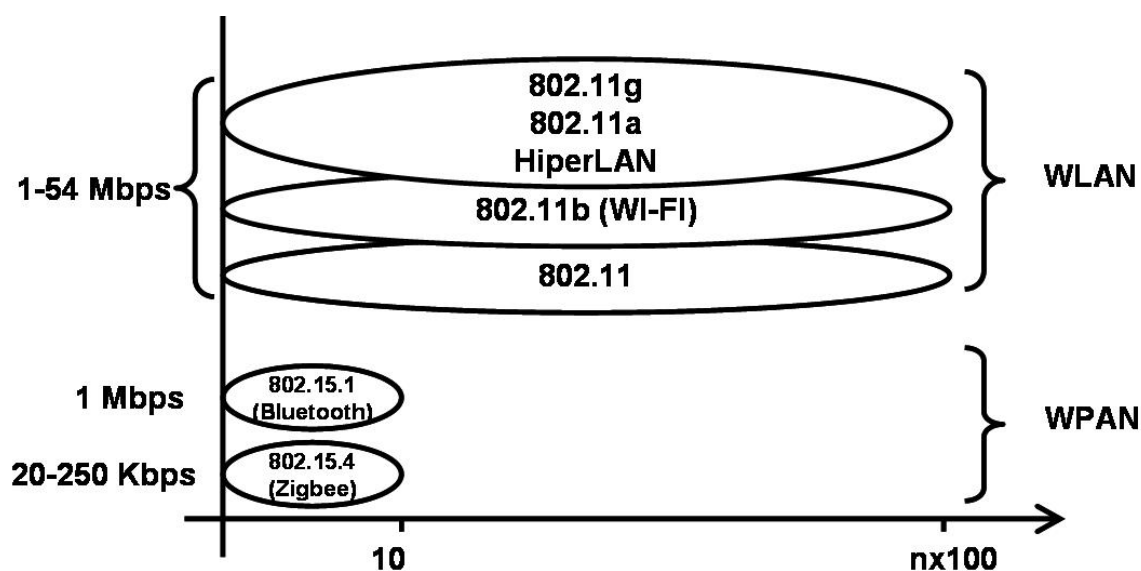


Figura 8. Comparativa de anchos de banda de las distintas tecnologías en bandas ISM más comunes

Por otro lado existen las redes WiFi, basadas en las especificaciones IEEE 802.11 y que pueden alcanzar una velocidad de decenas de Mbps. Los dispositivos de estas redes deben permanecer activos casi de forma permanente y permiten el intercambio de gran cantidad de datos, y por consiguiente, tienen consumos mucho más elevados.

Existen otros protocolos de carácter propietario, como por ejemplo el MiWi de Microchip Technology, que permite la creación de redes tipo *mesh* (igual que ZigBee), y que aunque con algunas carencias en cuanto al números de saltos, seguridad y número máximo de nodos en una misma red, que pueden ser utilizadas en sistemas de control. No obstante, para esta primera versión de sistema, se ha seleccionado la tecnología ZigBee básicamente por 3 motivos:

- Flexibilidad y compatibilidad con otros fabricantes
- Bajo *time-to-market*
- Estandarización de protocolos

II.2. Análisis Energético

En tecnología ZigBee se puede hablar de un consumo energético asimétrico, ya que depende de la función que el nodo realice dentro de la red. Para nodos enrutadores y coordinadores (tipos COO y "R" en la figura "Esquema general de los componentes del sistema"), el consumo es fijo, puesto que el dispositivo permanece activo el 100% del tiempo, recibiendo paquetes procedentes de otros nodos y transmitiendo información a la red, tanto propia como procedente de otros dispositivos. Este tipo de nodos tienen latencias muy bajas, permitiendo un flujo de comunicación de datos constante. Como contra partida, deben ir alimentados por baterías de alta capacidad o conectados a fuentes de alimentación externas (ejemplo, convertidor AC/DC desde 230VAC), ya que en el caso de un módulo tipo ETRX2, el consumo entre 40 y 50mA constantes.

Sin embargo, los nodos finales (tipo "F") solamente permanecen activos un corto periodo de tiempo, lo suficiente para preguntar a su "padre", el nodo enrutador mas cercano, o con el que tiene conexión (no tiene porque ser el mas cercano físicamente), si existe algún dato para él procedente de otro nodo de la red, o para entregar información del propio nodo final con destinatario otro u otros nodos. Este proceso es muy rápido, del orden de milisegundos, y la periodicidad con la que se realiza condiciona la latencia del sistema a la hora de ejecutar ordenes (actuaciones) o realizar peticiones de datos.

Las siguiente gráfica muestra los consumos para un mensaje de petición de información (mensaje tipo *poll*) desde un nodo final "F" a un nodo enrutador "R" o coordinador "COO".

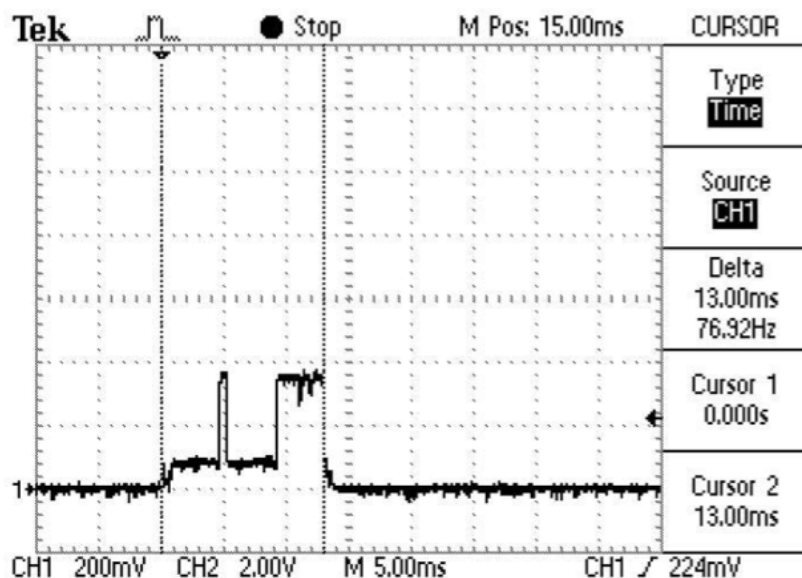


Figura 9. Coste energético para un mensaje de *poll* con o sin transferencia de datos desde nodo enrutador (medida en mV equivalente a $I_{cc} \times 10\Omega$).

Por tanto la frecuencia de los mensajes de *poll* influye directamente en el consumo de batería, existiendo una relación inversa (no lineal) entre la latencia y el consumo.

Otro hecho a tener en cuenta en el consumo de los nodos tipo "F" (finales) son los mensajes de envío de información sobre localización y estado de las entradas. Este tipo de mensajes supone un coste energético mayor que los mensajes de *poll*, debiendo reducirse al mínimo posible su uso, y ajustándose a los requisitos de cada aplicación en concreto (el sistema desarrollado en este TFG permite seleccionar los tiempos en función de la criticidad temporal de la señal a leer).

A continuación se indican el coste energético aproximado de cada tipo de mensaje medido de forma experimental:

Tabla 1. Coste energético (equivalente mA) por tipo de mensaje ($V_{cc} = 3.3v$)

Tipo de Mensaje	Coste energético equivalente en mA
<i>Poll</i>	0.22
<i>Track</i>	0.3
Lectura Entradas	1

La alimentación de los nodos finales “F” del sistema se realiza con una pila de litio primario tipo CR123A con una capacidad nominal de 1500mAh.

A continuación se muestra una comparativa de la duración de la batería estimada en función de la frecuencia de lectura de entradas, envío de *tracks* y mensajes de *poll*. El consumo de los posibles sensores conectados a la batería no se ha considerado. Se añade un consumo base del nodo de 2μA.

Tabla 2. Estimación de duración de batería tipo CR123A en función de los tiempos de transmisión

Tipo de Mensaje	Tiempos de transmisión (en segundos)		
<i>Poll</i>	1	3	3
<i>Track</i>			10
Lectura Entradas	3	5	
Duración Estimada	6 meses	1 año y 2 meses	2 años
Distribución Energética	<p>Consumo Base 0,47% Lectura Entradas 47,39% Poll 52,13%</p>	<p>Consumo Base 1,02% Lectura Entradas 61,43% Poll 37,54%</p>	<p>Consumo Base 1,90% Track (RUN) 28,48% Poll 69,62%</p>

II.3. Descripción del sistema:

II.3.1. Unidad Coordinadora

La unidad coordinadora es un elemento único en el sistema del cual depende la lógica de actuación del mismo, así como gran parte de las comunicaciones entre los distintos elementos (elemento COO en la figura “Esquema general de los componentes del sistema”).

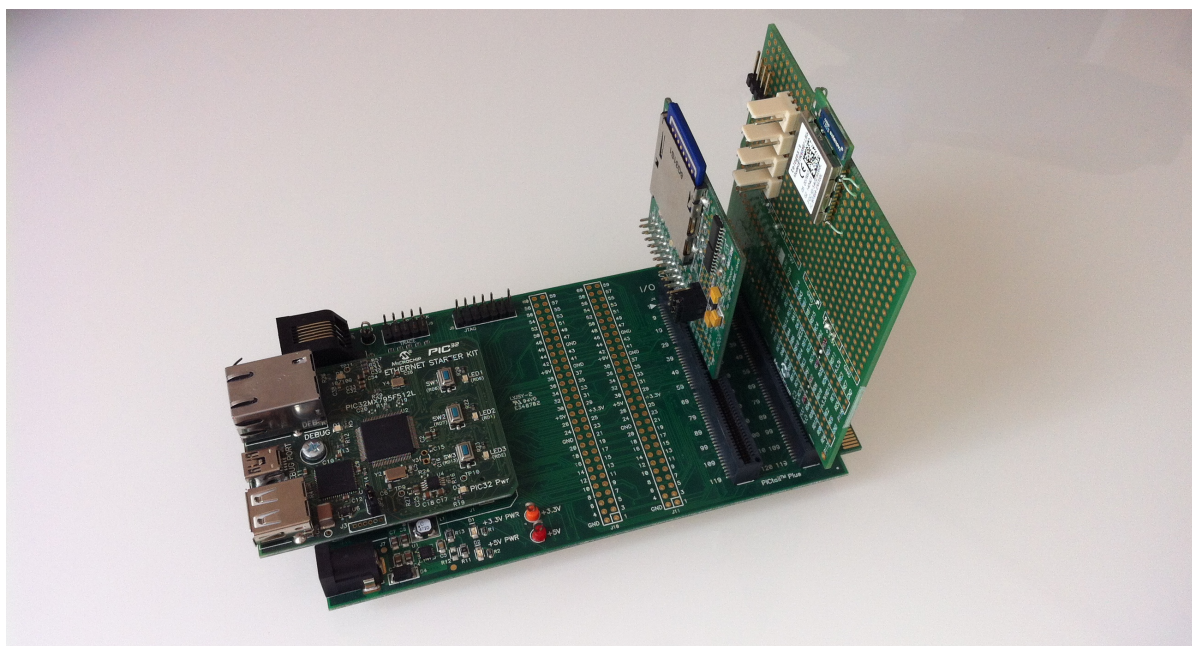


Figura 10. Prototipo de unidad coordinadora utilizada durante el desarrollo

a) Hardware:

El núcleo principal de procesamiento está basado en un procesador de 32 bits (microcontrolador), al cual se le han añadido periféricos externos que proporcionan, principalmente, vías de comunicación y actuación con el resto de elementos que forman el sistema.

Con el fin de ahorrar tiempo en el desarrollo así como reducir los costes del mismo, se ha utilizado una tarjeta de evaluación del fabricante Microchip. El modelo es denominado “PIC32 Ethernet Starter Kit” (Referencia: DM320004) y está orientado al desarrollo de aplicaciones que requieran comunicaciones a través de Ethernet (red LAN). Además proporciona puertos de comunicaciones USB tanto en los modos host, device y OTG, en principio, no utilizados para esta aplicación.

A continuación se resumen las principales características de esta tarjeta:

+ Núcleo controlador PIC32MX795F512L:

- Microcontrolador de 32 bits con núcleo M4K capaz de trabajar hasta 80Mhz.
- 512KB de memoria flash
- 12KB de memoria para bootloader
- 128KB de memoria RAM

- 8 canales DMA independientes con controlador HW
- Interfaz Ethernet MAC tipo MII/RMII
- Programador directo integrado (no requiere de interfaz externo entre PC de desarrollo y microcontrolador)



Figura 11. Tarjeta PIC32 Ethernet Starter Kit

- + Interfaz de control de capa física Ethernet y conector RJ45 con transformador de acoplamiento integrado.
- + Conector de expansión para actuar como interfaz con la aplicación (en este caso con la tarjeta expansora tipo “Starter Kit IO Expansion Board”).
- + Conexión directa a puerto USB para programación/depuración usando protocolo ICSP (propietario de Microchip). Elimina la necesidad de utilizar un programador/depurador externo.

En conjunto con el “PIC32 Ethernet Starter Kit”, se integra una tarjeta de expansión denominada “Starter Kit IO Expansion Board” (Referencia: DM320002). Esta tarjeta proporciona la expandibilidad necesaria al núcleo procesador de forma que puedan integrarse en el mismo los dispositivos externos, que en el caso particular del desarrollo de este sistema, serán módulos de comunicaciones radio y un lector para tarjetas de memoria tipo SD.

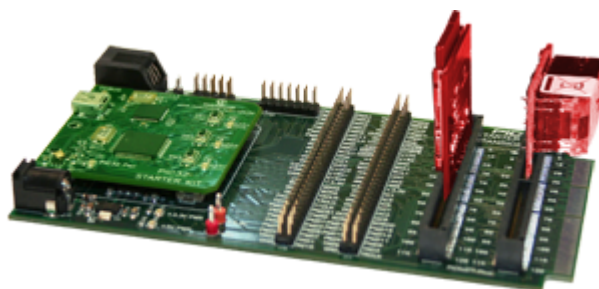


Figura 12. Tarjeta de expansión con núcleo procesador y elementos expansores integrados

Además, la tarjeta expansora proporciona alimentación tanto al núcleo como al resto de los elementos montados sobre ella a 5 y 3,3 voltios, lo cual posibilita de forma directa la integración de prácticamente cualquier módulo radio del mercado.

Sobre la tarjeta expansora, se han integrado los siguientes elementos, los cuales están directamente relacionados con el tipo de aplicación a implementar, ya que son los encargados de proporcionar, entre otras cosas, conectividad inalámbrica.

+ Módulo de comunicaciones ETRX3: Fabricado por Telegesis (Referencia: ETRX3587-LRS), y basado en el núcleo de Ember 357, este módulo integra todo lo necesario para dotar tanto a la unidad coordinadora como al resto de los elementos del sistema de conectividad inalámbrica tipo ZigBee (basado en el IEEE 802.15.4). En particular, implementa la pila denominada ZigBee Pro, que experimenta numerosas mejoras en comparación de sus predecesoras, sobretodo desde el punto de vista de expandibilidad y seguridad de la red. Este módulo seleccionado es del tipo ETRX357-LRS, el cual incluye un amplificador adicional que proporciona una potencia de salida máxima de hasta +20dBm (100mW), regulable por software.

Por otro lado, este tipo de módulos “todo en uno” facilitan la integración de la tecnología con terceras partes, en este caso, el núcleo procesador principal de la unidad coordinadora, el PIC32MX795F512L. La comunicación entre ambos se realiza mediante una vía serie (UART) configurada a 115200bps para una mayor velocidad de transferencia. El protocolo es propietario del fabricante (Telegesis) y está basado en comandos AT. La versión de FW en el ETRX3 utilizada durante todo el desarrollo es la R308X, la cual puede ser actualizada manteniendo su compatibilidad en cuanto a comandos y formato de datos se refiere.



Figura 13. Módulo de comunicaciones ZigBee ETRX3

Incluye hasta 24 pines de entrada/salida de propósito general, de los cuales 4 pueden utilizarse como entradas analógicas con un ADC de 14 bits y una referencia de tensión de 1.2v.

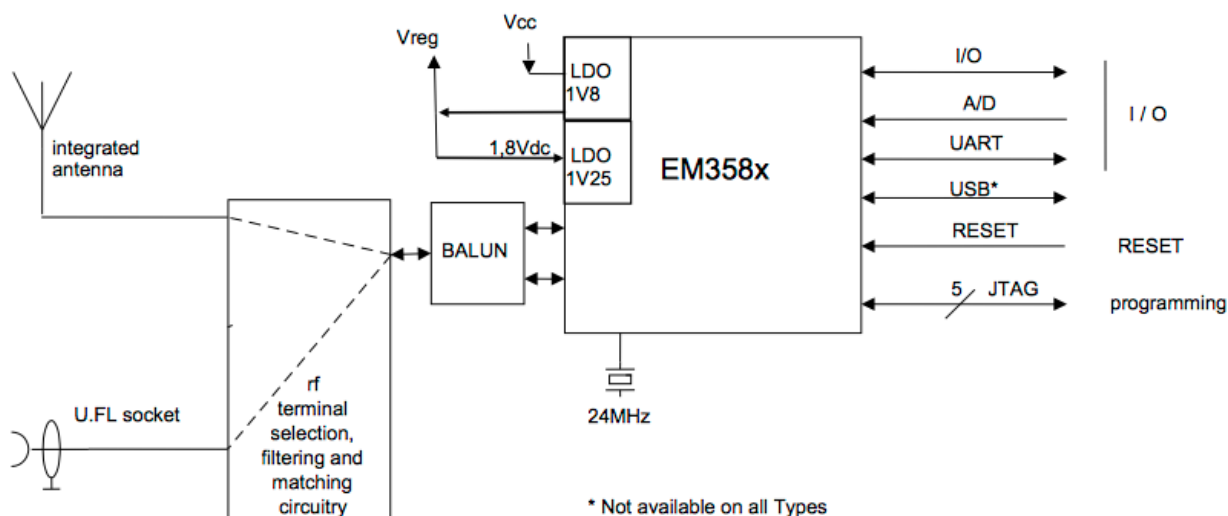


Figura 14. Esquema de bloques interno del módulo radio ETRX3

Como dispositivo de almacenamiento de configuración (y otros datos) del sistema, se ha habilitado una tarjeta SD extraíble. Esta configuración permite almacenar una gran cantidad de datos en la tarjeta (incluso archivos de *logs* en aplicaciones futuras) que pueden ser transferidos de forma sencilla a un dispositivo externo (ej. ordenador). Además, permite la actualización del firmware de la unidad coordinadora mediante bootloader, copiando la imagen del mismo en la tarjeta SD (no implementado en la actual versión).

La tarjeta SD está contenida en una PCB que puede ser acoplada directamente en la tarjeta de expansión. Se denomina “PICtail Board for SD & MMC”, también del fabricante Microchip (Referencia: AC164122).



Figura 15. *PICtail* almacenamiento de información en tarjetas SD

b) Software:

El software que ejecuta el núcleo de la unidad coordinadora ha sido desarrollado, en su mayor parte, utilizando lenguaje C compilado mediante XC32, un compilador específico para esta arquitectura de microprocesador que optimiza tanto la velocidad de ejecución como el tamaño de memoria ocupado.

Por otro lado, aproximadamente un 30% de la memoria de programa está ocupada por código en lenguaje HTML. Este lenguaje es interpretado por cualquier navegador Web, y es utilizado como

medio de transferencia de información entre el interfaz gráfico de configuración y visualización de datos utilizado por el usuario, y la memoria y lógica internas de la aplicación. No obstante, el microprocesador genera código HTML dinámico en función de las necesidades del interfaz gráfico de usuario.

El uso de HTML como interfaz de usuario proporciona gran flexibilidad de acceso al dispositivo, evitando la instalación de aplicaciones dedicadas o el desarrollo de un interfaz hardware (pantalla y botones/teclado en la propia unidad). Además posibilita el acceso remoto a través de internet con total transparencia tanto para la unidad coordinadora como para el usuario.

También existen rutinas internas programadas en lenguaje ensamblador. Estas son utilizadas para la escritura en registros críticos de sistema o para la ejecución de código con alta precisión temporal.

A continuación, se muestran los bloques principales de la unidad coordinadora y la relación y conexión entre los mismos:

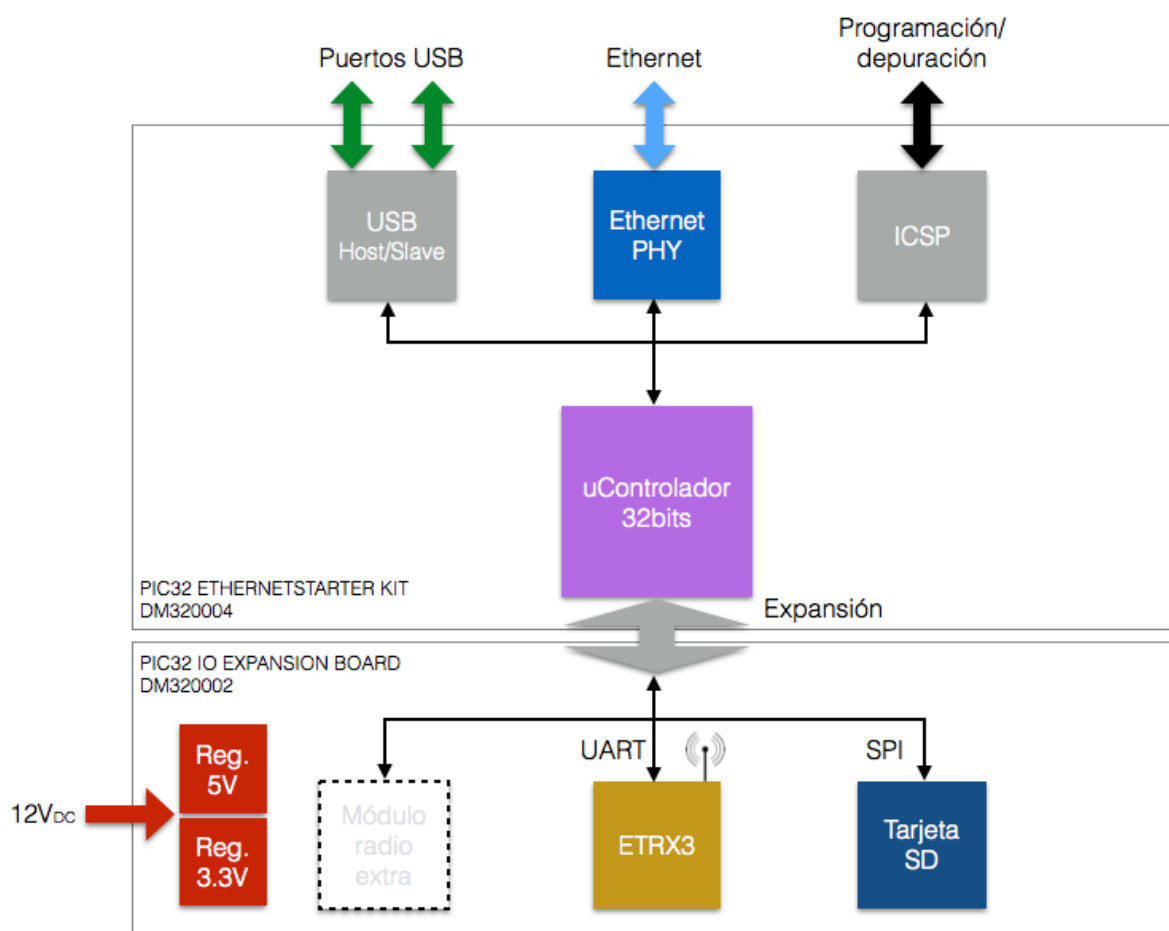


Figura 16. Bloques internos de la unidad coordinadora

II.3.2. Nodos inalámbricos

Los nodos inalámbricos, distribuidos a lo largo del escenario bajo monitorización y control, son los encargados de enviar datos sobre sensores a la unidad coordinadora y recibir órdenes de actuación.

desde esta. Disponen de una serie de puertos de entrada y/o salida, encargados de leer los datos procedentes de los sensores (entrada), y excitar los actuadores externos (salida).

Estos dispositivos están marcados como “R” Y “F” en la figura “Esquema general de los componentes del sistema”.

a) Nodos inalámbricos ZigBee

Este tipo de nodo está basado en el protocolo ZigBee y la pila IEEE 802.15.4. Trabajan en la banda ISM de 2.4Ghz y generalmente están disponibles en distintas versiones con potencias de transmisión que oscilan entre 0 y +20dBm. La sensibilidad del receptor suele ser común, en torno a los -118dBm.

En estos dispositivos se distinguen 2 tipos diferenciados, nodos finales y nodos enrutadores:

- Nodos finales (“F”): actúan como puntos finales de envío y recepción de información, es decir, no permiten el paso de los datos de la red inalámbrica a través de ellos como elemento de “salto”. Pueden ser alimentados con baterías, paneles solares de baja potencia, etc., ya que pueden ser configurados como dispositivos de ultra bajo consumo, llegando a necesitar corrientes de funcionamiento del orden de pocos μA a 3.3V.
- Nodos intermedios o enrutadores “: actúan, además de “como nodos finales F”, como dispositivos de enrutado y organización de red inalámbrica. Disponen de las mismas entradas y salidas que un nodo final y también pueden realizar funciones de monitorización y actuación. Son capaces de retransmitir tramas de datos de la red WSN, permitiendo la distribución de los mismos hacia los nodos finales. Son utilizados para extender la red a lugares “lejanos” (ejemplo, sin visión directa con la unidad principal coordinadora). Por otro lado, necesitan mayor energía que los nodos finales para su funcionamiento (en torno a 50-60mA), por lo que generalmente son alimentados desde 230V_{AC} o sistemas de alimentación ininterrumpida con baterías recargables.

En el desarrollo de este TFG se han utilizado las referencias ETRX2 y ETRX3 del fabricante Telegesis, las cuales proporcionan acceso a redes WSN ZigBee Pro, así como entradas y salidas tanto digitales como analógicas.

Como prototipos de desarrollo y pruebas, se han montado módulos con la siguiente configuración:

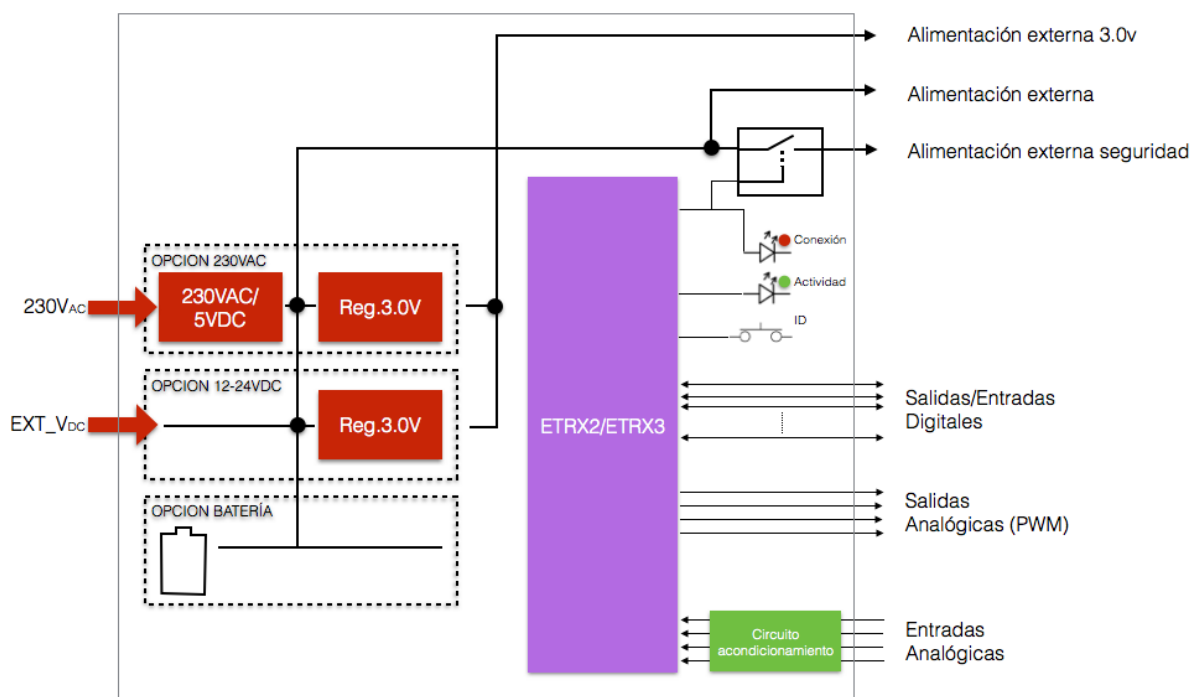


Figura 17. Bloques internos de los nodos inalámbricos ZigBee basados en el módulo ETRX2 y ETRX3

A continuación se describen las conexiones del módulo y sus funciones:

- 12/20 entradas/salidas digitales (ETRX2/ETRX3): Las salidas proporcionan una señal TTL para la activación de elementos externos (ej. relés, optoacopladores, LEDs indicadores, etc.). La máxima corriente será de 5mA por salida. Las entradas detectarán niveles alto y bajo TTL. Pueden ser utilizadas para la detección de cierre de circuitos (ej. finales de carrera) o conexión a sensores con detección de umbral.
- 1 salidas analógicas por PWM (ETRX2/ETRX3): Pueden conectarse a circuitos externos de regulación analógica (ej. circuitos de control de iluminación, controladores de motores, etc.).
- 2/4 entradas analógicas (ETRX2/ETRX3): Entradas analógicas de alta impedancia para la conexión de sensores externos. El rango de tensión de entrada será configurado según el tipo de sensor utilizado en cada caso. Las configuraciones más comunes serán las siguientes: 0-1.0v, 0-2v, 0-5v, 0-10v.
- Salidas de alimentación para elementos externos: El nodo es capaz de proporcionar alimentación a los sensores externos conectados a él. Para ello cuenta con 3 salidas diferenciadas:
 - Alimentación externa regulada: Proporciona una tensión regulada de 3.0VDC para la alimentación de sensores que requieran una tensión fija y estable. Solo está disponible con alimentación desde 230VAC y EXT_VDC (ver "Tabla 4: Compatibilidades y tipos de salida de alimentación en función de la fuente de alimentación de entrada").
 - Alimentación externa: Proporciona una tensión continua para la alimentación de sensores externos. En caso de alimentación tipo 230VAC, esta será de 5VDC (regulada). En caso de EXT_VDC o Batería, la tensión proporcionada será la misma alimentación de entrada externa o batería (ver "Tabla 4: Compatibilidades y tipos de salida de alimentación en función de la fuente de alimentación de entrada").

- Alimentación externa seguridad: Con las mismas características que la salida de “Alimentación externa”, esta salida solo permanecerá activa mientras el módulo se encuentra conectado a la red WSN. En caso contrario, se eliminará la tensión mediante el uso de una *switch* de “lado alto”. Esto permite utilizar el nodo para el control de dispositivos de seguridad, de los cuales se debe tener conocimiento (*feedback*) de su estado para un correcto funcionamiento, ya que el nodo podría quedar fuera de red debido a elementos externos (ej. pérdida de cobertura radio) mientras alguno de sus actuadores permanece activo. De no usar este tipo de salida para la alimentación del actuador, la actuación permanecería activa permanentemente. Esta alimentación no estará disponible hasta al menos 10 segundos después de la conexión del nodo a la red WSN o la alimentación del mismo.
- El nodo inalámbrico admite 3 tipos de alimentación: 230VAC, 12VDC y alimentación por pila o batería. Según el tipo de aplicación, se montarán los componentes específicos de cada una de las 3 opciones (no coexistentes).
 - 230VAC (compatible con nodos tipo “F” y “R”): Esta versión incorpora una fuente de alimentación capaz de proporcionar 1A a 5VDC, la cual puede ser utilizada para la alimentación de elementos externos (sensores y/o actuadores). A su vez se integra un regulador que proporciona al módulo una tensión de 3.0VDC para su funcionamiento interno.
 - EXT_VDC (compatible con nodos tipo “F” y “R”): Esta opción de alimentación admite una tensión de entrada entre 5 y 12VDC. Puede ser utilizada para la alimentación del módulo desde baterías externas y/o placas solares, baterías de vehículos, alimentadores externos domésticos, etc. La propia tensión de entrada también está disponible para la alimentación de sensores/actuadores externos.
 - Batería (compatible con nodos tipo “F”): En esta opción, el módulo incorpora un portapilas para la inserción de una batería de litio primaria (no recargable) o secundaria (recargable, aunque esta versión inicial de nodo no tiene la capacidad de hacerlo). El tamaño de batería será tipo CR123A (H 34.5mm, Ø 17mm). Este tipo de baterías, ampliamente utilizadas en el mercado y de bajo coste, proporcionan una tensión nominal de 3.0VDC y cuentan con una capacidad estándar de 1500mAh.

La siguiente tabla resume las compatibilidades del nodo en función de su fuente de alimentación con respecto a las salidas de alimentación externas y el tipo de nodo.

Tabla 3. Compatibilidades y tipos de salida de alimentación en función de la fuente de alimentación de entrada

	230VAC	EXT_VDC	BATERÍA/PILA
Alimentación externa regulada 3.0VDC	SI	SI	NO
Alimentación externa	SI, 5VDC regulada	SI, EXT_VDC	SI, tensión batería
Alimentación externa seguridad	SI, 5VDC regulada	SI, EXT_VDC	SI, tensión batería
Compatible tipo “F”	SI	SI	SI
Compatible tipo “R”	SI	SI	NO

Además, el nodo incorpora 2 LEDs de indicación de actividad que permiten conocer *insitu* si el nodo está conectado a red y dispone de alimentación. También cuenta con un pulsador de identificación que permite al nodo ser identificado en el interfaz de configuración Web a través de la unidad coordinadora.

Tabla 4. Estados y funciones de LEDs indicadores

	LED VERDE	LED ROJO
Apagado	Nodo sin alimentación	Nodo con conexión a WSN
Parpadeo	Nodo activo (alimentado)	Conexión a WSN perdida o no establecida

b) Nodos inalámbricos propietarios

El sistema ha sido diseñado para la coexistencia de varias tecnologías radio sobre una misma plataforma, tanto desde el punto de vista *software* como *hardware*. Aunque durante el desarrollo de este TFG se están realizando trabajos en este sentido, no estarán finalizados en el momento de la entrega del mismo. Ver sección “II.8. Trabajos Futuros” para más información.

II.3.3. Sensores y Actuadores

Los sensores y actuadores son elementos externos, totalmente dependientes de la aplicación, que son integrados con el sistema.

Existen en el mercado multitud de sensores para la medición, por ejemplo, de parámetros ambientales como presión, temperatura, humedad relativa, etc. y que pueden ser integrados directamente en los nodos inalámbricos a través de sus entradas en analógicas.

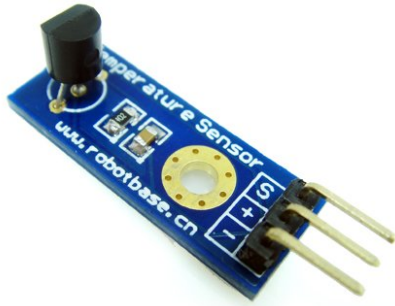
La salida de estos, normalmente, suele cumplir los estándares de mercado en tensión o corriente (ejemplo, salidas 0.4-2.0v o 4-20mA), además existen en el mercado multitud de adaptadores que proporcionan una salida según estos estándares para sensores resistivos (ej. PT100 para aplicaciones industriales) entre otros.

Alimentación	12-24VDC
Entrada	Transductor tipo PT-100
Salida	Analógica 4-20mA



Figura 18. Adaptador 4-20mA para sensores de temperatura tipo PT100

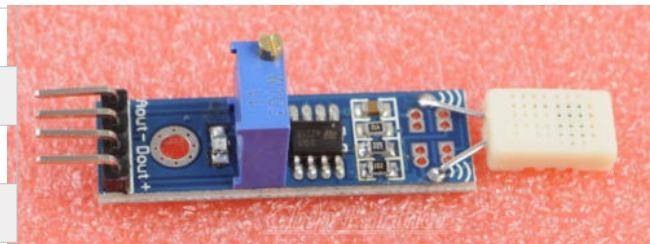
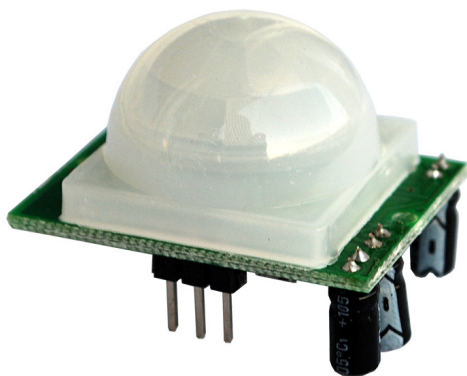
No obstante, también existe la posibilidad de integrar sensores chip en las unidades inalámbricas. Por ejemplo, la referencia LM35 es un sensor de temperatura de bajo coste y ultra bajo consumo con una precisión de 1°C que puede ser directamente conectado a una de las entradas analógicas. A continuación se muestran algunos sensores que pueden adquirirse en el mercado a muy bajo coste (entre 2 y 3€) para aplicaciones domésticas:



Alimentación	4-30 VDC
Consumo	<60μA
Salida	Tensión 10mV/°C
Rango	0-150°C (>0v)

Figura 19. Sensor de temperatura basado en LM35

Alimentación	3.3-5 VDC
Consumo	<20μA
Salida	Tensión 0-2V
Rango	0-100 %HR

Figura 20. Sensor de humedad basado en el amplificador operacional LM393 mas *hygristor* HR31

Alimentación	3.3-5 VDC
Consumo	<50μA
Salida	TTL
Rango	1-5 metros

Figura 21. HC-SR501: Sensor de presencia PIR

Alimentación	5 VDC
Consumo	150mA (calentador)
Salida	Resistiva (1-20K)
Rango	300 - 10000ppm

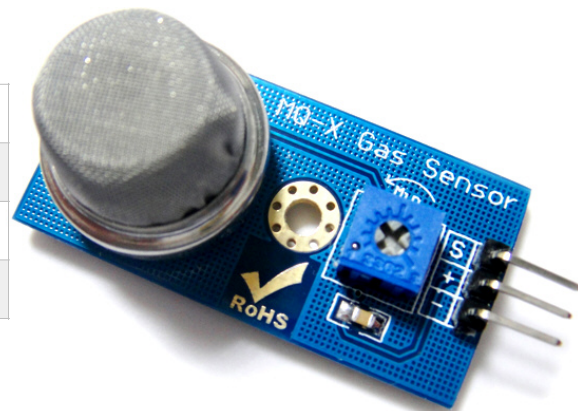
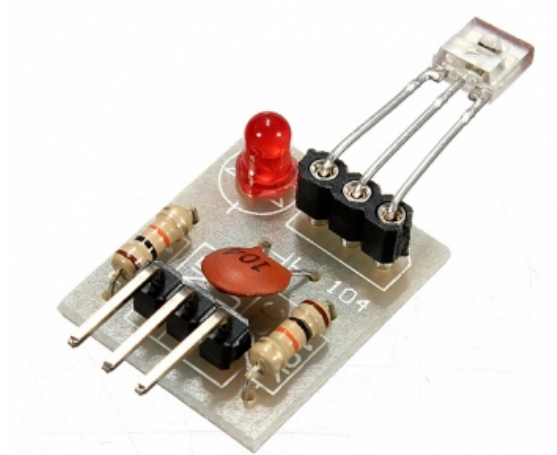


Figura 22. MQ-2: Sensor de gas butano/metano



Alimentación	5 VDC
Consumo	<1mA
Salida	TTL

Figura 23. Módulo de receptor láser (ej. receptor para barrera láser)

En cuanto a actuadores se refiere, el sistema permite la conexión, entre otros, de módulos de relés y transistores. Con estos dispositivos de actuación, se cubren las demandas de actuación mas comunes, ya que permiten la actuación tanto analógica en corriente continua (transistor con PWM) como digital en corriente continua y alterna (relé).

Alimentación	5 VDC
Consumo bobina	150mA
Contactos	10A @ 250VAC

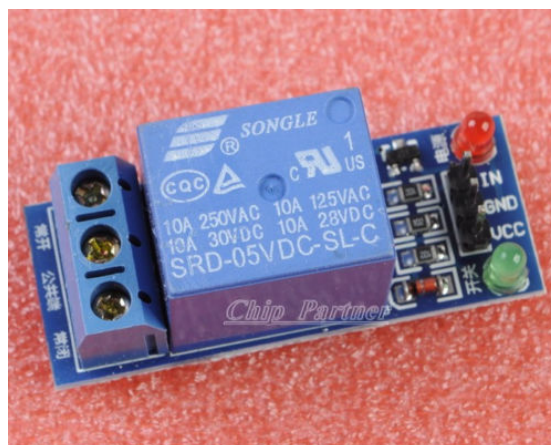


Figura 24. Módulo de actuación mediante relé

II.3.4. Terminales de visualización

El sistema cuenta (opcionalmente) con terminales de visualización y control en los que se muestra el estado de los parámetros bajo monitorización. A través de estos terminales pueden visualizarse los datos recibidos desde los nodos remotos, incluyendo las lecturas de los sensores conectados a ellos y otros datos relevantes como tensión de batería e indicadores de actividad (indican si el nodo remoto está "vivo" o no).

Los terminales de visualización están basados en elementos externos al sistema, tabletas electrónicas de bajo coste o dispositivos móviles, que no forman parte del desarrollo del mismo.



Figura 25. Ejemplo de tableta básica de bajo coste utilizada como terminal de visualización

El uso de este tipo de dispositivos y un interfaz de visualización basado en Web (HTML) como terminal de visualización, en lugar de un sistema propietario (ej. pantalla conectada a la unidad coordinadora), proporciona al sistema las siguientes ventajas:

- Terminales de bajo coste, disponibles en el mercado por menos de 50€.
- Uso de terminales “ilimitados”. La unidad coordinadora dispone de un máximo de conexiones admisibles. Este número puede elevarse, pero no infinito.
- Uso remoto. A través una red local, enlace WiFi o Internet, los terminales de visualización pueden estar situados en lugares remotos, lejos del escenario bajo monitorización y con total transparencia para el usuario.

II.4. Funciones del sistema e interfaz de configuración

El núcleo de procesamiento lógico del sistema se encarga de tomar las decisiones de actuación. Los procesos implicados están totalmente independizados de la capa física, es decir, en esta parte de procesamiento, el tipo de nodo inalámbrico es transparente, ya que las comunicaciones son gestionadas en otro “hilo”. Esto permite utilizar un único núcleo lógico con varios tipos de interfaces físicos. El enlace entre ambos bloques son las denominadas GPDAIOs (*General Purpose Digital-Analog Input-Output*), cada una de ellas asignada a un nodo en particular y con unas funciones específicas (entrada analógica, salida analógica PWM, entrada digital y salida digital).

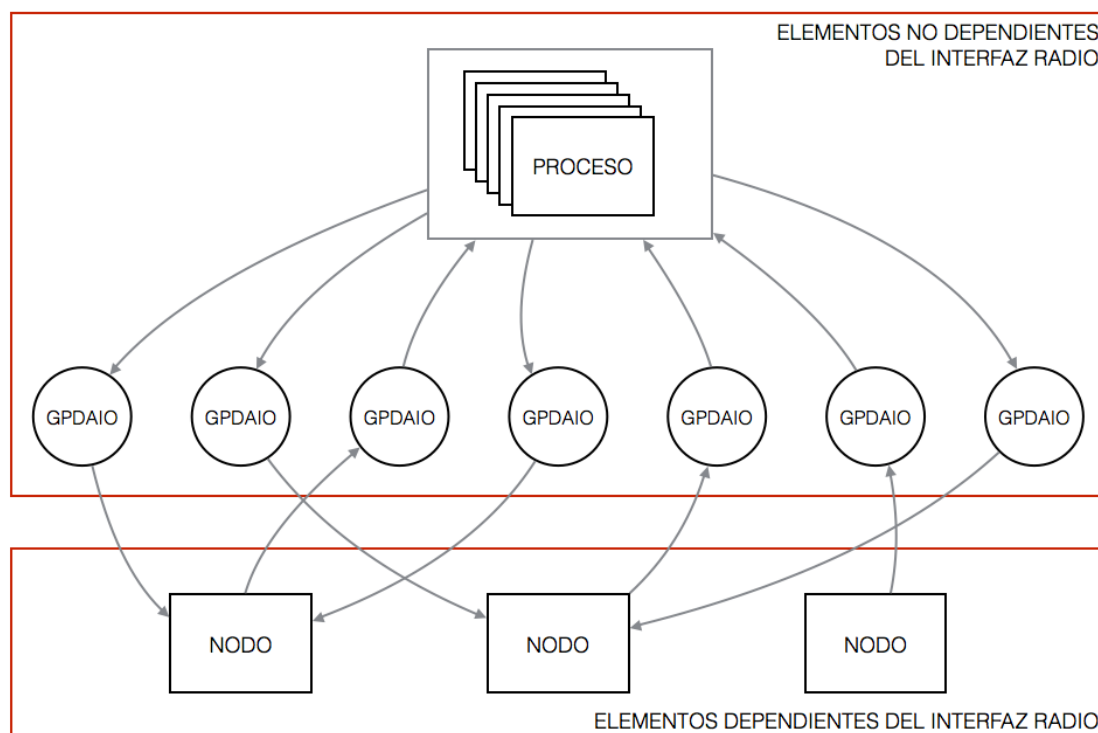


Figura 26. Objetos a nivel de procesamiento. Los GPDAIOs se utilizan como enlace entre los nodos (dependientes del HW) y los procesos (genéricos)

El sistema es capaz de actuar sobre sus salidas (GPDAIOs) en función de los valores leídos en sus entradas (GPDAIOs), estados temporales, estado de otros procesos, interacción del usuario, etc. Esta lógica de actuación está configurada únicamente en la unidad coordinadora, no existiendo la posibilidad de que un nodo tome decisiones por sí mismo por muy sencilla que sea la lógica. Por tanto el sistema siempre requiere de la unidad coordinadora para su funcionamiento.

II.4.1. Procesos

La configuración de los estados de las salidas en función de los parámetros de entrada se realiza a través del interfaz gráfico de usuario Web, mediante la parametrización de máquinas de estados llamadas “procesos”. Cada proceso es ejecutado 1 vez por segundo, y está constituido por 4 estados:

- **READY (listo):** Normalmente utilizado para indicar el estado de reposo, es decir, que todo está listo esperando la señal de puesta en marcha. (ejemplo, el nivel de agua en un depósito es suficiente como para garantizar presión en el sistema de riego, solo hay que accionar la electroválvula para comenzar).

- **RUNNING** (en funcionamiento): Indica que el proceso está activo. (ejemplo, el sistema de riego está en marcha).
- **STOP** (parada): El proceso se ha detenido. (ejemplo, el tiempo de riego ha expirado o el nivel de depósito del agua destinada al riego es bajo).
- **BLOCKED** (bloqueo): El proceso no puede ejecutarse y queda bloqueado. Destinado a indicar errores de funcionamiento o situaciones anómalas. (ejemplo, no hay datos del sensor de nivel de agua).

Las transiciones entre procesos se realizan de forma cíclica entre READY, RUNNING y STOP, y únicamente este sentido. Esto imposibilita que el sistema pueda pasar de un estado READY a un estado STOP de forma directa sin realizar una transición a través de RUNNING.

Sin embargo, para el estado BLOCKED se permite una transición directa desde READY y STOP, y una indirecta (pasando por STOP desde el estado RUNNING). Esto permite que en caso de existir alguna anomalía, el sistema bloquee el proceso en cualquier momento.

La no disponibilidad de datos de cualquier GPDAIO (ejemplo, pérdida de comunicación con el nodo inalámbrico) implicada en el proceso supondrá una transición a estado BLOCKED.

La siguiente figura muestra un resumen de las transiciones entre los distintos estados del proceso:

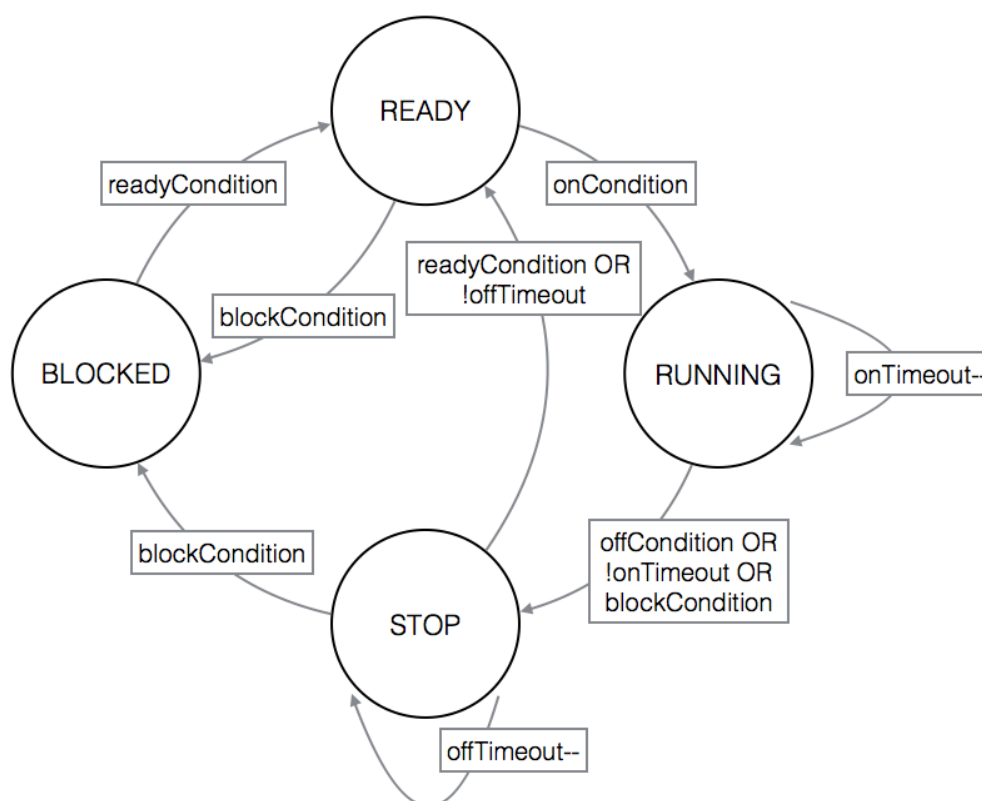


Figura 27. Máquina de estados con transiciones entre procesos y principales condiciones de cambio

El sistema, según se ha desarrollado en este TFG, permite un total de 32 procesos. No obstante con una optimización del uso de memoria RAM del procesador o ampliación de la misma (puede ampliarse con memorias externas) el número de procesos puede incrementarse significativamente.

II.4.2. Transiciones o condiciones de cambio

Dentro de un proceso existen condiciones de cambio (transiciones) entre los distintos estados del mismo. Para que una transición se lleva a cabo, deben cumplirse ciertas condiciones, parametrizadas por el usuario a través del interfaz Web.

Dicha parametrización toma como valores de entrada tanto entradas del sistema como salidas y otros parámetros internos. La combinación entre ellos puede agruparse en distintos niveles (grupos de operaciones) y mediante operadores AND y OR, dentro de cada uno de los estados de un proceso y de forma totalmente independiente entre ellos (ver sección “II.5.1.c. Configuración de procesos”). Una vez procesados todos los niveles del estado de un proceso, el resultado debe ser *true* (1), para que se realice una transición y consecuentemente, el estado de las salidas cambie al determinado (también configurable por el usuario).

El procesamiento de las condiciones para cada uno de los estados se realiza según el siguiente esquema simplificado, mediante el uso de llamadas recursivas. El número de niveles máximos se ha limitado a 3.

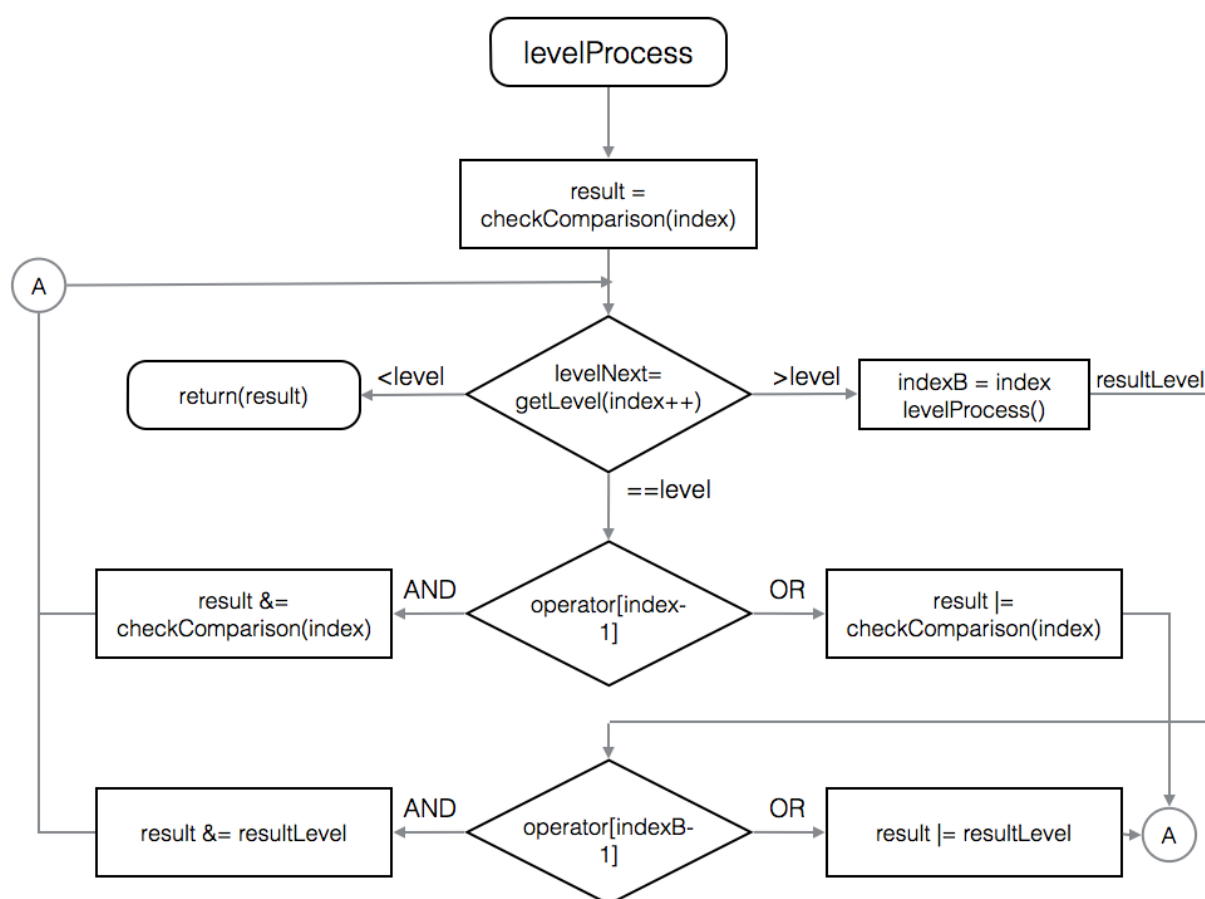


Figura 28. Procesamiento de los niveles de condición dentro de un estado

Las condiciones para realizar el cambio a un determinado estado vienen determinadas por el tipo de elementos a comparar (ejemplo, para una entrada digital solo serán válidos las condiciones distinto de, <>, o igual que, ==).

La siguiente tabla resume el tipo de condición según el tipo de entrada/salida:

Tabla 5. Resumen de tipos de factores permitidos como entradas de un estado, tipo de condición y factores de comparación permitidos.

Factor A	Condiciones permitidas	Factor B permitido
Entrada Digital	<> ==	Entrada Digital Salida Digital Valores: ON, OFF
Salida Digital	<> ==	Entrada Digital Salida Digital Valores: ON, OFF
Entrada Analógica	<> == > < >= <=	Entrada Analógica Salida Analógica (PWM) Valores: rango factor B
Salida Analógica	<> == > < >= <=	Entrada Analógica Salida Analógica (PWM) Valores: rango factor B
<i>Time Out</i> (contador de tiempo)		Rango: 10 segundos - 1 semana
Fecha y hora	<> == > < >= <=	Fecha y hora indicada
Presencia	<> ==	Nodo inalámbrico con transmisión de presencia (<i>track</i>) activado.
<i>Always</i>		

- Entrada digital: cualquier GPDAIO tipo *Digital Input*.
- Salida digital: cualquier GPDAIO tipo *Digital Output*.
- Entrada analógica: cualquier GPDAIO tipo *Analog Input*.
- Salida analógica: cualquier GPDAIO tipo *PWM Output*.
- *Time Out*: contador de cuenta atrás con la duración especificada en el Factor B.
- Fecha y Hora: Fecha y hora determinadas en el factor B. En caso de no indicar alguno de los campos, se considerará repetitivo (ejemplo, si solamente se indica hora, minuto y segundo, a la condición se le presupone una repetibilidad diaria).

- Presencia: indica si debe existir o no, la presencia de un nodo en particular en la zona de cobertura con transmisión de mensajes de localización (*track*) activos.
- *Always*: La condición de entrada al estado siempre será verdadera.

II.4.3. Estado de las salidas

De igual forma, las salidas (GPDAIOs) son actualizadas en cada transición según los parámetros específicos de cada estado. Configurables por el usuario, el tipo de asignación depende del tipo de salida seleccionada.

Tabla 6. Factores y condiciones permitidas según el tipo de salida seleccionada

Factor A	Condiciones permitidas	Factor B permitido
Salida Digital	\diamond $==$	Entrada Digital Salida Digital Valores: ON, OFF
Salida Analógica PWM	$.==$	Entrada Analógica Valores: 0-100

II.5. Servidor Web

La configuración y parametrización del sistema se realiza íntegramente desde un interfaz Web, ya que el firmware del microcontrolador incorpora un servidor HTTP con su correspondiente pila TCP/IP. Esto permite que el usuario acceda a la configuración y visualización de datos desde cualquier punto de la red a la que sistema esté conectado, incluso remotamente a través de internet.

El servidor Web se ha implementado con generación dinámica de código HTML, así como procesamiento de formularios Web tipo POST y GET, lo cual flexibiliza el interfaz, permitiendo la representación de datos en tiempo real en pantalla en función de la página a la que se acceda en cada momento y sin necesidad de actualizar la misma.

El interfaz está diseñado para facilitar al usuario las tareas de configuración, de forma que, por ejemplo, al seleccionar un tipo de entrada en el factor A de la comparación en un proceso, la página Web se actualiza automáticamente y sólo muestra el tipo de entradas o salidas compatibles en el factor B, así como ciertas opciones de condición.

La programación del interfaz se ha realizado tanto en HTML y JavaScript, como en C para la generación de código dinámico y la representación de variables en tiempo real. Un hecho a tener en cuenta en este sentido, y debido a la utilización de un núcleo con "bajos recursos" como para actuar de servidor Web de páginas complejas y dinámicas, ha sido el tamaño de los buffers de entrada y salida. En cada petición desde la Web (ejemplo, mediante el envío de un formulario por método POST), el software comprueba que se ha recibido toda la información necesaria antes de comenzar el procesamiento, puesto que esta puede ser recibida en paquetes distintos, en función de la memoria disponible en cada momento. De igual forma, al enviar información, esta es ajustada al tamaño del buffer de salida disponible en cada momento en el interfaz Ethernet. En caso de necesitar enviar o recibir datos adicionales, el proceso libera al microcontrolador y realiza una llamada posterior para continuar con el envío/recepción de información.

A continuación se muestra un ejemplo de rutina no bloqueante para el envío de fragmentos HTML dinámico con comprobación de tamaño del buffer de memoria de salida disponible (ver sección "IV. Software" para mas información.

```
// Rellena el desplegable de seleccion de terminal de visualizacion
void HTTPPrint_vtSelect(void){
    WORD len;
    static BYTE cadena[128], *pCadena;
    static UINT16 vtIndex;

    // Inicializacion de datos en la primera llamada
    if(curHTTP.callbackPos == 0u)
    {
        curHTTP.callbackPos = 1; // Se pondra a cero cuando hayan salido todos los datos
        cadena[0]=0;
        pCadena = cadena;
        vtIndex = 0;
    }

    // Determine how many bytes we can write
    len = TCPIsPutReady(sktHTTP);

    while(len && curHTTP.callbackPos)
    {
        if(!(*pCadena)) // Si el puntero no esta apuntando a ningún sitio, se le asigna
                        una nueva cadena que transmitir
        {
            if(VT[vtIndex].name[0]==0) // No tiene nombre
                sprintf((char*)cadena, "<option value=\"%u\">New Terminal</option>", vtIndex);
            else
                sprintf((char*)cadena, "<option value=\"%u\">%s</option>", vtIndex,
                        VT[vtIndex].name);

            vtIndex++;
            pCadena = cadena;
        }
    }
}
```



```

    }

    while(len && *pCadena) // Mientras existan datos que enviar y hueco para ponerlos,
                           // pues se envían
        len -= TCPPut(sktHTTP, (BYTE)*pCadena++);

    // Si ha terminado de transmitir la cadena y el índice ha llegado al final, esta función
    // no se llama más.
    if(!(*pCadena) && vtIndex==MAX_VT_NUMBER)
        curHTTP.callbackPos=0;
    }
}

```

Figura 29. Ejemplo de rutina de envío de código dinámico HTML en función de los terminales de visualización configurados

El servidor Web necesita la reserva de memoria RAM para cada una de las conexiones entrantes (sockets abiertos por equipos remotos). Estas se han limitado a 8 con el fin de evitar problemas de asignación de memoria para los procesos internos o problemas con la pila (ubicada al final de la memoria RAM). El buffer de transmisión y recepción de datos desde la Web (ejemplo, envío de formularios POST y GET), también se ha limitado a 256 bytes, suficiente para procesar todos los formularios contenidos en el interfaz de configuración.

II.5.1. Interfaz de configuración

El interfaz de configuración está formado por una estructura de páginas de Web que permiten la configuración y parametrización completa del sistema. Algunas de ellas están protegidas por contraseña. Los *cookies* instalados en el navegador almacenan la información de seguridad (nombre de usuario y contraseña), evitando que el usuario tenga que introducirlos cada vez que acceda.

La estructura de las páginas se muestra en la siguiente figura:

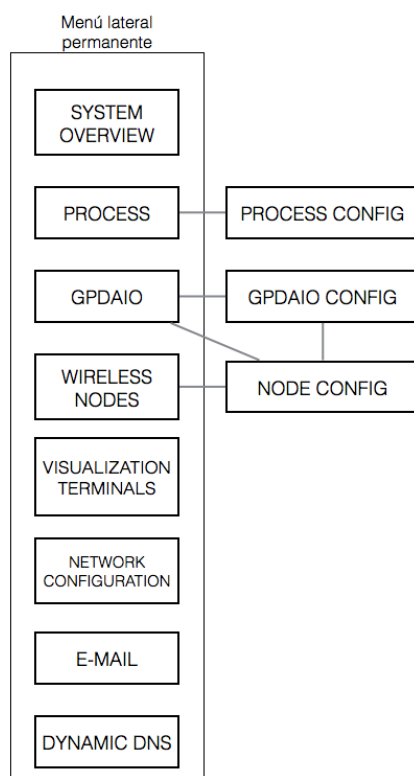


Figura 30. Páginas Web de configuración y enlaces internos

a) System Overview:

Muestra información básica sobre el sistema, tales como el número de nodos inalámbricos configurados, número de GPDAIOs, la versión de la pila TCP/IP utilizada y la fecha de compilación de la versión de firmware que corre en la unidad coordinadora. Además, indica la fecha y hora configurada y permite su sincronización con el ordenador remoto desde el que se accede a la Web.

TFG_FJ_ESPADA

System Overview	Wireless Monitoring System Overview
Process	System RTCC: 13:09:38 16/08/2014
GPDAIO	Active Wireless Nodes: 8
Wireless Nodes	Active GPDAIOs: 3
Visualization Terminals	Stack Version: v5.42
Network Configuration	Build Date: Sep 15 2014 00:08:00
E-mail	Update System RTCC with local computer parameters: 13:09:49 16/08/2014
Dynamic DNS	

Figura 31. Página de información general del sistema

b) Process

Muestra el listado de procesos activos y permite el acceso a la configuración de procesos adicionales. El listado incluye información básica del proceso, como la prioridad del mismo, nombre asignado y estado actual en tiempo real.

Si un proceso se encuentra configurado, el enlace "Edit" permite la modificación de los parámetros del mismo, mientras que si el proceso se encuentra sin configurar el enlace "Config" nos llevará a la Web de configuración con todos los parámetros vacíos.

La columna de la derecha, permite eliminar uno o varios procesos activando los *checks* correspondientes. No se podrá eliminar un proceso que no está configurado. Una vez seleccionados, el botón "Save" eliminará los procesos y guardará los datos en la memoria SD.



TFG_FJ_ESPADA

System Overview

Process

GPDAIO

Wireless Nodes

Visualization

Terminals

Network Configuration

E-mail

Dynamic DNS

Process List

P	Name	Status	Config	Del
1	ProcesoCiclico	RUNNING	Edit	<input type="checkbox"/>
2	o1	READY	Edit	<input type="checkbox"/>
3	Cambio_cada_10_Seg	READY	Edit	<input type="checkbox"/>
4	RIEGO	READY	Edit	<input type="checkbox"/>
5	DEPOSITO	READY	Edit	<input type="checkbox"/>
6	ALARMA	READY	Edit	<input type="checkbox"/>
7	TEST_TEMPORIZACION	READY	Edit	<input type="checkbox"/>
8	-	-	Config	<input type="checkbox"/>
9	-	-	Config	<input type="checkbox"/>
10	-	-	Config	<input type="checkbox"/>
11	-	-	Config	<input type="checkbox"/>
12	-	-	Config	<input type="checkbox"/>
13	-	-	Config	<input type="checkbox"/>
14	-	-	Config	<input type="checkbox"/>
15	-	-	Config	<input type="checkbox"/>
16	-	-	Config	<input type="checkbox"/>

Trabajo Final de Grado - Francisco José Espada Moreno - Sept. 2014

Figura 32. Web con el listado de procesos activos en el sistema

c) Process Config

Esta sección permite la configuración de los procesos según la lógica deseada por el usuario. En la parte superior se muestra un desplegable desde el cual el usuario puede seleccionar el estado del proceso a configurar.

Una vez seleccionado, aparece la configuración actual del estado, en caso de existir. En caso contrario, la configuración aparecerá en blanco.

El sistema permite asignar un nombre a un estado dado (campo *Name*), de forma que el usuario pueda identificarlo más fácilmente en los terminales de visualización.

A continuación, aparecen 2 tablas en las que se seleccionan tanto las condiciones de entrada del estado como los niveles en las salidas de actuación mientras el sistema permanece en el mismo. En el caso de las entradas, estas se mostrarán según la compatibilidad de tipo y condiciones mostradas en la tabla "Resumen de tipos de factores permitidos como entradas de un estado, tipo de condición y factores de comparación permitidos". Las flechas laterales situadas en la columna de la parte izquierda, sirven para incrementar o decrementar el nivel de operación (agrupar condiciones). El sistema actual permite hasta 3 niveles de operación y 6 condiciones por estado.

Las salidas permanecerán según la configuración del estado durante toda la duración del mismo. En caso de asignación de salidas a entradas, estas se modificarán en tiempo real mientras el estado permanezca activo.

El botón "Save" permite guardar los cambios realizados.

PROCESS CONFIG

Process name:

Status:

READY STATUS

☒ Send e-mail notification when input condition

INPUT CONDITION

<	>	PuertaAbierta	==	Value	OFF	OR
<	>	SensorHumedadA	==	Value	20.000000 HR	OR
<	>	TemperaturaSalon	>	Value	20.000000 C	AND
<	>	PuertaAbierta	==	Value	ON	
<	>	Select Factor A		Value		
<	>	Select Factor A		Value		

OUTPUT STATUS

Activacion_Alarm	=	ON
Select Output	=	
Select Output	=	
Select Output	=	
Select Output	=	

Trabajo Final de Grado - Francisco José Espada Moreno - Sept. 2014

Figura 33. Página de configuración de procesos y estados del los mismos

d) GPDAIO

En esta página se muestra un listado con todas las GPDAIOs configuradas en el sistema. En las columnas de la tabla se indican el nombre asignado a la misma, el tipo de GPDAIO (entrada digital,

entrada analógica, etc.), el nodo asignado que la contiene y el estado de la misma en tiempo real ("ON", "OFF", etc.).

Existen 2 enlaces en cada un de las filas de la tabla. Haciendo click sobre el nombre del nodo, aparecerá la página de configuración de dicho nodo en particular, mientras que el enlace "Config" permite abrir la página de configuración de dicha GPDAIO en concreto.

No está permitido añadir ni eliminar GDAIOs desde esta página, esta función es específica de cada nodo y debe realizarse desde la página NODE CONFIG.



TFG_FJ_ESPADA

System Overview

Process

GPDAIO

Wireless Nodes

Visualization
TerminalsNetwork
Configuration

E-mail

Dynamic DNS

General Purpose Digital-Analog Input-Output List

N	Name	Value	Type	Node	Config
1	TemperaturaSalon	0.0C	Analog Input	Nodo de pruebas	Edit
2	VentiladorGrande	OFF	Digital Output	Nodo de pruebas	Edit
3	Activacion_Alarm	OFF	Digital Output	NodoCesped	Edit
4	PuertaAbierta	OFF	Digital Input	Nodo de pruebas	Edit
5	SensorHumedadA	0.0HR	Analog Input	NodoCesped	Edit
6	ActivacionBomba	OFF	Digital Output	NodoCesped	Edit

Trabajo Final de Grado - Francisco José Espada Moreno - Sept. 2014

Figura 34. Web con el listado de GPDAIOs configuradas en el sistema

e) GPDAIO Configuration

Permite configurar las entradas y salidas de un nodo según los tipos permitidos por el hardware correspondiente.

El desplegable de la parte superior permite seleccionar el tipo de entrada y/o salida permitido por el hardware del nodo inalámbrico en cuestión. Los tipos de entradas más comunes (y en principio, soportadas por el sistema) son:

- Entrada Analógica: permite convertir unidades de entrada (tensión, corriente, etc.) en el parámetro físico a medir. Esto se realiza mediante la parametrización por parte del usuario de los valores máximo y mínimo de cada una de las dos unidades, por ejemplo, un sensor de 0.4 a 2.0 voltios (valores mínimo y máximo de la señal de entrada), que realiza una medida de temperatura de 0 a 150°C (valores mínimo y máximo del parámetro físico a medir). Además, permite indicar la unidad de la magnitud física a monitorizar (grados centígrados, metros por segundo, etc.). El dato convertido y la unidad se muestran tanto en los terminales de visualización como en las Webs de configuración de procesos y estado de entradas.
- Entrada Digital: permite seleccionar si la entrada tiene lógica directa (ON=nivel alto, OFF=nivel bajo), o invertida.
- Salida Analógica PWM: no permite ninguna parametrización. Para este tipo de salida, los valores irán referidos a porcentaje (de 0 a 100%), correspondiente al ciclo de trabajo de la señal PWM.

- Salida Digital: de igual modo que la Entrada Digital, puede configurarse como directa o inversa.

Todas las entradas permiten la asignación de un nombre (ejemplo, "Temperatura cocina"), que será mostrado en los terminales de visualización y Webs de configuración.

El botón "Save", almacena los cambios realizados en la configuración.



Hello, SensorHumedadA
TFG_FJ_ESPADA

System Overview

Process

GPDAIO

Wireless Nodes

Visualization Terminals

Network Configuration

E-mail

Dynamic DNS

General Purpose Digital-Analog Input-Output Configuration

GENERAL PARAMETERS

Name:

Type:

Node:

ANALOG INPUT PARAMETERS

Measured Units:

Min: **Max:**

Signal Input (V):

Min: **Max:**

Trabajo Final de Grado - Francisco José Espada Moreno - Sept. 2014

Figura 35. Página de configuración de parámetros de GPDAIOs

f) Wireless Nodes

Lista los nodos existentes en el sistema, así como información básica de cada uno de ellos.

El listado muestra el nombre asignado al nodo, el tipo del mismo (ejemplo, ETRX2), el estado actual (*ONLINE* / *OFFLINE*), nivel de señal recibida y *time stamp* de la última recepción (cuándo se han recibido datos del nodo por última vez).

Esta Web es sólo informativa y está destinada a realizar diagnósticos del sistema en caso de fallo. La edición de los parámetros de los nodos se realiza a través del enlace "Config" situado en la parte derecha de cada fila de la tabla, el cual lleva a la Web NODE CONFIG.



System Overview		Wireless Node List						
Process								
GPDAIO								
Wireless Nodes								
Visualization Terminals								
Network Configuration								
E-mail								
Dynamic DNS								
N	Type	Name	Status	RSSI	Time Stamp	Config	Del	
1	ETRX2	PROTO1	ONLINE	0	2014/09/10 12:25:56	Config	<input type="checkbox"/>	
2	ETRX2	PROTO2	ONLINE	0	2014/09/10 12:25:56	Config	<input type="checkbox"/>	
3	ETRX2	TEST_BATERIA	ONLINE	0	2014/09/10 12:25:56	Config	<input type="checkbox"/>	
4	ETRX2	DEPOSITO	ONLINE	0	2014/09/10 12:25:56	Config	<input type="checkbox"/>	
5	ETRX2	ALARMA	ONLINE	0	2014/09/10 12:25:56	Config	<input type="checkbox"/>	
6	ETRX2	NodoCesped	ONLINE	0	2014/09/10 12:25:56	Config	<input type="checkbox"/>	
7	ETRX2	Imaginario_test	ONLINE	0	2014/09/10 12:25:56	Config	<input type="checkbox"/>	
8	ETRX2	Nodo de pruebas	ONLINE	0	2014/09/10 12:25:56	Config	<input type="checkbox"/>	
Save								

Trabajo Final de Grado - Francisco José Espada Moreno - Sept. 2014

Figura 36. Web con listado de nodos inalámbricos configurados en el sistema

g) Wireless Node Configuration

Permite configurar un nodo inalámbrico con los parámetros correspondientes al tipo de nodo. Por ejemplo, en el caso de un nodo ZigBee tipo ETRX2/3, los parámetros a configurar son Nombre (parámetro común a cualquier tipo de nodo), tiempo de transmisión (cada cuanto tiempo el nodo envía datos sobre el estado de sus entradas a la unidad coordinadora), nodo final o router ("F" o "R" según la figura "Esquema general de los componentes del sistema") y localizador (el nodo emitirá mensajes de posición para habilitar la función de localización).

También permite configurar las entradas y salidas (GPDAIOs) del nodo según las funciones admitidas por cada una de ellas (ejemplo, la IO3 de un nodo tipo ETRX2 puede ser configurada como Entrada Digital, Salida Digital o Salida Analógica PWM). Los tipos admitidos se indican en la parte de la izquierda de la tabla. El identificador de la GPDAIO en el nodo coincide con el indicado en la primera columna (izquierda). Además, muestra el nombre asignado para cada entrada, así como el tipo.

El enlace "Edit" (disponible para las GPDAIOs ya configuradas), permite modificar los parámetros de configuración de cada GPDAIO, mientras que el enlace "Config", permite añadir una nueva GPDAIO al nodo y al sistema.



System Overview
Process
GPDAIO
Wireless Nodes
Visualization Terminals
Network Configuration
E-mail
Dynamic DNS

Wireless Node Configuration

Select Node:

Node Type: ETRX2			
Node Name: <input type="text" value="Nodo de pruebas"/>			
Node MAC: 1122334455667788			
Parent MAC: 0000000000000000			
Sink MAC: 0000000000000000			
Link Quality: 0			
RSSI: 0 dB			

N	DI	DO	PWM	ANA	Name	Type	Config	Del
IO0	•	•		-		-	Config	<input type="checkbox"/>
IO1	•	•		-		-	Config	<input type="checkbox"/>
IO2	•	•		-		-	Config	<input type="checkbox"/>
IO3	•	•	•	-		-	Config	<input type="checkbox"/>
IO4	•	•		-		-	Config	<input type="checkbox"/>
IO5	•	•			PuertaAbierta	Digital Input	Edit	<input type="checkbox"/>
IO6	•	•		-		-	Config	<input type="checkbox"/>
IO7	•	•		-		-	Config	<input type="checkbox"/>
IO8	•	•		-		-	Config	<input type="checkbox"/>
IO9	•	•		-		-	Config	<input type="checkbox"/>
IO10	•	•			VentiladorGrande	Digital Output	Edit	<input type="checkbox"/>
IO11	•	•		-		-	Config	<input type="checkbox"/>
AN0			•		TemperaturaSalon	Analog Input	Edit	<input type="checkbox"/>
AN1			•	-		-	Config	<input type="checkbox"/>

☐ Delete node and associated GPDAIOs

Trabajo Final de Grado - Francisco José Espada Moreno - Sept. 2014

Figura 37. Web de configuración de parámetros de nodos inalámbricos

h) Visualization Terminals

Esta Web permite configurar los datos mostrados en los terminales de visualización.

El desplegable de la parte superior permite seleccionar el terminal correspondiente. Actualmente el sistema permite configurar hasta un máximo de 8 terminales de visualización distintos (en los que se muestran distintos datos, pero a los que tienen acceso varios dispositivos remotos al mismo tiempo). Si el terminal dispone de nombre asignado, este será mostrado en el desplegable. En caso contrario aparecerá el nombre por defecto "Visualization Terminal X".

Una vez seleccionado, los parámetros configurados en el mismo (si existen) aparecerán en la tabla de configuración situada en la parte inferior de la página, mientras que el aspecto final del terminal se mostrará en la parte media.

La tabla de configuración permite indicar el nombre del terminal, mostrado en la parte superior del mismo (ejemplo, "Control Jardín"), y cada uno de los datos a mostrar en los indicadores de la parte superior e inferior.

Los indicadores de la parte superior permiten mostrar el estado de entradas, salidas y procesos de forma extendida, es decir, en los procesos, se indicará el nombre asignado por el usuario al estado

actual del mismo. Además, disponen en la parte derecha de un botón de actuación configurable en los procesos. En los indicadores de la parte inferior, se muestran los mismos datos pero de modo simplificado y menor tamaño. En este caso, el nombre de los estados de un proceso no son mostrados, mientras que los estados con su configuración genérica, BLOCKED, READY, RUNNING y STOP, se muestran con indicadores rojo, azul, verde y naranja.

En la parte inferior del terminal aparecen accesos al resto de los terminales configurados.

El botón "Save" permite almacenar los cambios realizados.



TFG_FJ_ESPADA

- System Overview
- Process
- GPDAIO
- Wireless Nodes
- Visualization Terminals
- Network Configuration
- E-mail
- Dynamic DNS

Visualization Terminal Configuration

Select Visualization Terminal:

Visualization Terminal Name:

Data refresh rate (seconds):

<input type="text" value="ActivacionBomba"/>	<input type="text" value="RIEGO"/>	<input type="text" value="SensorHumedadA"/>
<input type="text" value="Cambio_cada_10_Seg"/>	<input type="text" value="Activacion_Alarm"/>	
<input type="text" value="TemperaturaSalon"/>	<input type="text" value="ActivacionBomba"/>	
<input type="text" value="ProcesoCiclico"/>	<input type="text" value="DEPOSITO"/>	
<input type="text" value="-"/>	<input type="text" value="ActivacionBomba"/>	

Trabajo Final de Grado - Francisco José Espada Moreno - Sept. 2014

Figura 38. Web de configuración de datos a representar en los terminales de visualización

i) Network Configuration

Esta sección permite configurar los parámetros de red local de la unidad coordinadora.

Permite cambiar su dirección MAC (para fines experimentales), así como el *Host Name*, nombre asignado al sistema para su identificación en red, y que permite la coexistencia de varias unidades coordinadoras bajo una misma red local. El *Host Name* aparece en la parte superior derecha de todas las Web de configuración de una misma unidad coordinadora. Además, permite el acceso a dicha unidad a través de la siguiente dirección: **http://HostName**

El resto de parámetros a configurar son estándar en cualquier red IP (dirección IP, pasarela, máscara de red, servidor DNS primario y servidor DNS secundario). No obstante, cuando el dispositivo es conectado a un *router* con servidor DHCP, este puede asignar la configuración de red automáticamente mediante la activación del *check* "Enable DHCP"



~config_hostname~

System Overview
Process
GPDAIO
Wireless Nodes
Visualization Terminals
Network Configuration
E-mail
Dynamic DNS

Local Area Network Configuration

CAUTION: Incorrect settings may cause the system to lose network connectivity.

Enter the new settings for the board below:

MAC Address:	<input type="text" value="00:04:A3:13:0F:A3"/>
Host Name:	<input type="text" value="TFG_FJ_ESPADA"/>
	<input type="checkbox"/> Enable DHCP
IP Address:	<input type="text" value="192.168.1.52"/>
Gateway:	<input type="text" value="192.168.1.1"/>
Subnet Mask:	<input type="text" value="255.255.0.0"/>
Primary DNS:	<input type="text" value="192.168.1.1"/>
Secondary DNS:	<input type="text" value="0.0.0.0"/>
	<input type="button" value="Save Config"/>

Trabajo Final de Grado - Francisco José Espada Moreno - Sept. 2014

Figura 39. Web de configuración de los parámetros de red

j) E-Mail Notification

Configuración de los parámetros que permiten el envío de notificaciones vía e-mail.

Únicamente se debe configurar el servidor de correo saliente (tipo SMTP) con su dirección (IP o dominio), puerto (por defecto el 25), nombre de usuario y contraseña. En el campo "To" debe indicarse la dirección de correo del destinatario.

El envío de notificaciones por e-mail puede activarse en la Web *Process Config*. Las notificaciones se enviarán cuando un determinado proceso cambie a un determinado estado siempre y cuando se active el *check* "Send e-mail notification when input condition" de dicho estado.



~config_hostname~

System Overview

Process

GPDAIO

Wireless Nodes

Visualization Terminals

Network Configuration

E-mail

Dynamic DNS

E-Mail Nofitication

SMTP Server: **Port:**

User Name:

Password:

To:

Trabajo Final de Grado - Francisco José Espada Moreno - Sept. 2014

Figura 40. Web de configuración de los parámetros de correo saliente SMTP

k) Dynamic DNS

Permite la configuración de la unidad coordinadora para su acceso desde internet sin necesidad la contratación de una dirección IP fija.

El sistema admite la configuración de los siguientes servicios de DNS dinámicos (algunos de ellos gratuitos): DynDNS, No-IP y DNS-O-Matic.



~config_hostname~

System Overview

Process

GPDAIO

Wireless Nodes

Visualization Terminals

Network Configuration

E-mail

Dynamic DNS

Dynamic DNS

The current configuration is not valid.

DDNS Service:

User Name:

Password:

Host:

Trabajo Final de Grado - Francisco José Espada Moreno - Sept. 2014

Figura 41. Configuración de los parámetros para el uso de servicios de DNS dinámicos

II.6. Caso Práctico De Aplicación

Como ejemplo de aplicación y como fase final de validación, se propone una aplicación real en la que poner en práctica el sistema. Durante esta fase, se han utilizado los prototipos y el firmware desarrollados en este TFG.

La aplicación en particular está centrada en el control de los elementos de una finca rústica o casa de campo, ya que son un tipo de entornos que requieren bastante cuidado y atención, pero por otro lado, no son habitados de forma continua, por lo que a veces resulta complicado realizar un mantenimiento adecuado. Mediante el uso del sistema desarrollado se pretende automatizar parte de las labores, facilitando el mantenimiento de las instalaciones y mejorando la calidad de las mismas.

II.6.1. Escenario bajo monitorización y control

El sistema tiene como objetivo controlar y actuar sobre los elementos del siguiente escenario:

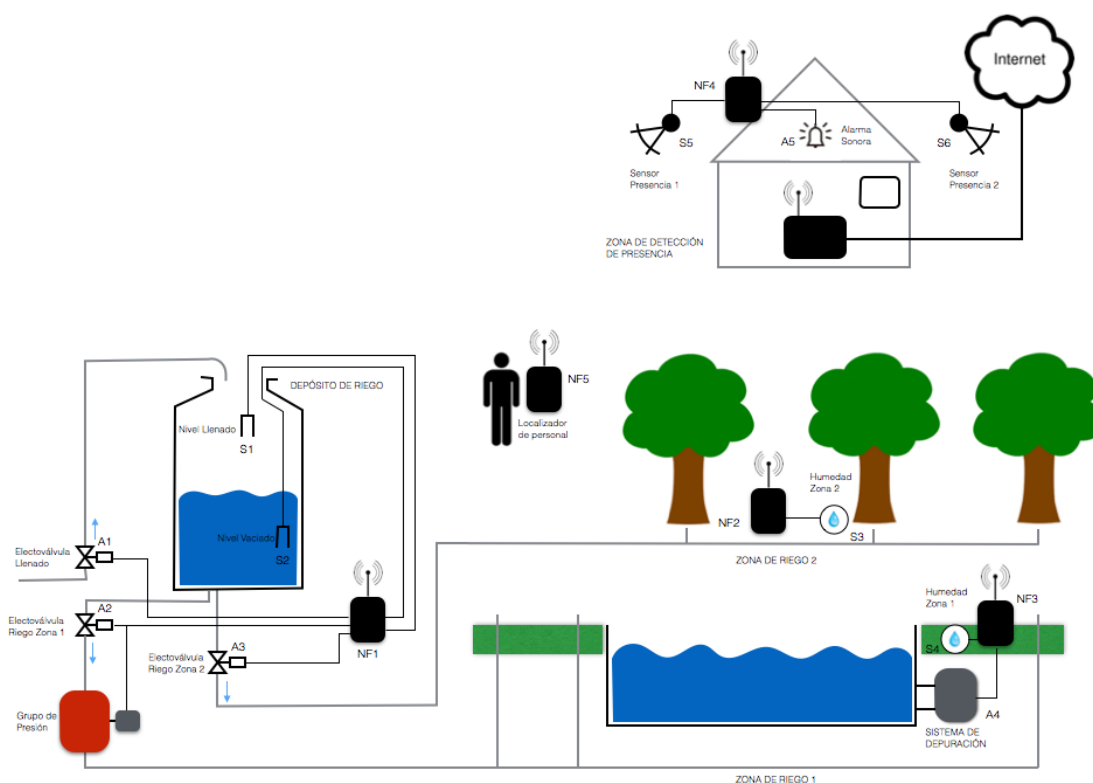


Figura 42. Escenario real de aplicación del sistema

II.6.2. Nodos inalámbricos del sistema

La configuración en cuanto a nodos inalámbricos y GPDAIOs del sistema es la siguiente:

Tabla 7. Listado de nodos inalámbricos configurados en el escenario bajo monitorización

LISTADO DE NODOS INALÁMBRICOS CONFIGURADOS EN EL SISTEMA							
Ref.	Nombre en sistema	Tipo de HW	R/F	Tx Datos (seg)	Poll (seg)	Loc. (seg)	Función
NF1	Depósito	ETRX2	R	10	1		Control de electroválvulas de riego y llenado de depósito
NF2	Frutales	ETRX2	F	30	5		Detección de humedad en zona de riego 2
NF3	Piscina	ETRX2	R	10	1		Detección de humedad en zona de riego 1 y activación de sistema de depuración
NF4	Alarma	ETRX2	R	10	1		Detección de presencia y activación de alarma
NF5	Localizador	ETRX2	F		5	10	Nodo personal de localización

II.6.3. Configuración de las GPDAIOs

La configuración de sensores y su conexión a las GPDAIOs correspondientes se detalla en la siguiente tabla:

Tabla 8. Listado de GPDAIOs configurados en el escenario bajo monitorización

LISTADO GPDAIOS CONFIGURADOS EN EL SISTEMA					
Ref.	Nombre	IO	Tipo	Nodo	Función
A1	Electro LLenado	IO0	Digital Output	Depósito	Electroválvula del control de llenado de depósito de agua
A2	Electro Césped	IO2	Digital Output	Depósito	Electroválvula de riego para zona 1 (césped piscina). A su vez, pone en funcionamiento el grupo de presión.
A3	Electro Frutales	IO3	Digital Output	Depósito	Electroválvula de riego para zona 2 (árboles frutales)
S1	LLenado superior	IO4	Digital Input	Depósito	Sensor de llenado de la parte superior del depósito de agua. Detecta cuando el depósito está lleno.
S2	LLenado inferior	IO5	Digital Input	Depósito	Sensor de llenado de la parte inferior del depósito de agua. Detecta cuando el depósito está vacío.
S3	Humedad Frutales	AN0	Analog Input	Frutales	Sensor de humedad de terreno en zona de riego 2 (frutales).
S4	Humedad Césped	AN0	Analog Input	Piscina	Sensor de humedad de terreno en zona de riego 1 (césped).
A4	Depuración	IO0	Digital Output	Piscina	Activación de sistema de depuración piscina.
A5	Alarma	IO0	Digital Output	Alarma	Activación de alarma acústica
S5	Presencia Izquierda	IO1	Digital Input	Alarma	Detección de presencia en lateral izquierdo de casa
S6	Presencia Derecha	IO2	Digital Input	Alarma	Detección de presencia en lateral derecho de casa

II.6.4. Procesos que intervienen en el sistema

Los procesos de control en función de las entradas y salidas configuradas se exponen a continuación:

a) Control Depósito

Depósito de riego (proceso CONTROL DEPÓSITO): Se trata de un depósito de agua destinada al riego alimentado desde una fuente externa intermitente, por lo que es necesario almacenar el agua para utilizarla durante las horas o días en los cuales la fuente externa no permite suministro. Para ello se han instalado 2 sensores de nivel de agua dentro del depósito (uno en la parte superior, S1 y otro en la parte inferior del mismo, S2) a través de los cuales el sistema es capaz de discriminar si el depósito está suficientemente vacío para comenzar el llenado o suficientemente lleno como para interrumpir el flujo de agua procedente del exterior.

Por otro lado, existe una electroválvula (A1) que controla el flujo del agua hacia el interior del depósito. Esta electroválvula permanecerá abierta (activa) durante el proceso de llenado e inactiva el resto del tiempo.

La salida de actuación del sistema estará alimentada mediante una salida de “Alimentación Externa de Seguridad” (ver “Tabla 4: Compatibilidades y tipos de salida de alimentación en función de la fuente de alimentación de entrada”). El sistema debe evitar a toda costa que el depósito se desborde, ya que provocaría daños en el entorno.

Aunque estos tres elementos (S1, S2 y A1) están conectados al mismo nodo, el sistema actuaría de la misma forma si estuviesen distribuidos en distintos nodos, ya que la capacidad de actuación (lógica del sistema) recae sobre la unidad coordinadora.

La siguiente tabla muestra las condiciones de entrada y estado de salida para cada una de las partes del proceso de control de depósito de riego:

Tabla 9. Condiciones de entrada y niveles de salida para cada uno de los estados del proceso de control de llenado de depósito

PROCESO: CONTROL DEPÓSITO						
Estado	Input/Output	Factor A	Con	Factor B	Op	Descripción
READY	INPUT	Always				La condición de entrada al estado READY siempre se cumple.
	OUTPUT	Electro LLenado	.=	OFF		
START	INPUT	LLenado Inferior	.=	OFF		Cuando el sensor de la parte inferior del depósito detecta que no existe nivel de agua.
	OUTPUT	Electro LLenado	.=	ON		Apertura de la electroválvula para iniciar el llenado
STOP	INPUT	LLenado Superior	.=	ON		El sensor de la parte superior del depósito detecta presencia de agua
	OUTPUT	Electro LLenado	.=	OFF		Cierre de la electroválvula para evitar el rebose del depósito
BLOCKED	INPUT					
	OUTPUT					

b) Riego Césped

Sistema de riego zona 1 (proceso RIEGO CÉSPED): Existen 2 zonas de riego independientes dentro del recinto, ambas con la misma filosofía de trabajo. La zona 1 está dedicada al riego del césped, la cual incorpora una bomba de presión hidráulica para el correcto funcionamiento del sistema de riego por aspersión.

Con el fin de optimizar el agua disponible, el sistema solamente debe activar el riego cuando sea necesario, es decir, cuando la humedad contenida en el terreno de la zona baje de cierto umbral. Esta lógica permite ahorrar agua cuando el terreno se moja por causas externas (días de lluvia).

Para ello, el sistema cuenta con un sensor de humedad relativa en la zona cubierta por el sistema de aspersión (S4, conectado al nodo NF3), el cual enviará información constante a la unidad coordinadora sobre el estado del terreno. Cuando el nivel de humedad baje de un determinado valor, el sistema de riego se activa mediante el actuador A2, conectado a NF1 (actúa conjuntamente sobre una electroválvula de paso y sobre la bomba de presión). A partir de este momento el riego permanecerá activo durante un periodo de tiempo de 1 hora (configurable desde la unidad de

control), puesto que el sensor de humedad quedará anegado de agua durante el proceso, y por tanto sus datos no son concluyentes para detener el sistema.

Tabla 10. Condiciones de entrada y niveles de salida para cada uno de los estados del proceso de control de riego de césped

PROCESO: RIEGO CÉSPED						
Estado	Input/Output	Factor A	Con	Factor B	Op	Descripción
READY	INPUT	LLenado Inferior	.=	ON		Debe existir al menos algo de agua en el depósito para poder iniciar el riego
	OUTPUT	Electro Césped	.=	OFF		Electroválvula cerrada
START	INPUT	<i>TimeOut</i>	.=	1 hora		El proceso de riego se detendrá 1 hora después de su inicio
	OUTPUT	Humedad Césped	<	50%		Cuando la humedad relativa del césped sea inferior al 50%, se inicia el proceso de riego
		Electro Césped	.=	ON		Apertura de la electroválvula para iniciar el riego
STOP	INPUT					No existen condiciones de entrada, al proceso anterior se la ha asignado un <i>TimeOut</i>
	OUTPUT	Electro Césped	.=	OFF		Electroválvula cerrada
BLOCKED	INPUT	LLenado Inferior	.=	OFF		Si no existe agua en el depósito, el proceso estará bloqueado
	OUTPUT	Electro Césped	.=	OFF		Electroválvula cerrada

c) Riego Frutales

Sistema de riego zona 2 (proceso RIEGO FRUTALES): Este sistema es idéntico al de la zona 1. El umbral de detección y el tiempo de riego entre zonas puede variar puesto que se trata de distintos tipos de vegetación y terreno, pero la filosofía de funcionamiento es la misma. En este caso la sensorización se realiza con el sensor S3 (conectado a NF2), mientras que la actuación sobre la electroválvula (no se requiere bomba de presión en este caso al tratarse de un sistema de riego por goteo) se realiza con A3 (conectado a NF1).

Tabla 11. Condiciones de entrada y niveles de salida para cada uno de los estados del proceso de control de riego de árboles frutales

PROCESO: RIEGO FRUTALES						
Estado	Input/Output	Factor A	Con	Factor B	Op	Descripción
READY	INPUT	LLenado Inferior	.=	ON		Debe existir al menos algo de agua en el depósito para poder iniciar el riego
	OUTPUT	Electro Frutales	.=	OFF		Electroválvula cerrada
START	INPUT	<i>TimeOut</i>	.=	1 hora		El proceso de riego se detendrá 1 hora después de su inicio
		Humedad Frutales	<	50%		Cuando la humedad relativa del terreno de los árboles frutales sea inferior al 50%, se inicia el proceso de riego
	OUTPUT	Electro Frutales	.=	ON		Apertura de la electroválvula para iniciar el riego
STOP	INPUT					No existen condiciones de entrada, al proceso anterior se la ha asignado un <i>TimeOut</i>
	OUTPUT	Electro Frutales	.=	OFF		Electroválvula cerrada
BLOCKED	INPUT	LLenado Inferior	.=	OFF		Si no existe agua en el depósito, el proceso estará bloqueado
	OUTPUT	Electro Frutales	.=	OFF		Electroválvula cerrada

d) Depuración

Sistema de depuración (proceso DEPURACIÓN): El sistema de depuración de la piscina también estará controlado por el sistema. Puesto que el agua requiere un tratamiento constante, esta deber ser depurada diariamente para la correcta distribución de los agentes tratantes (cloro, anti-algas, etc.).

Durante esta instalación de pruebas, el sistema estará encargado de activar la bomba de depuración de agua de forma periódica durante determinadas horas del día y durante una determinada época del año (programadas en el mismo), a través del actuador A4, conectado al nodo NF3.

A su vez, el usuario puede activar remotamente de forma manual el sistema de depuración con el fin

de prolongar el proceso cuando sea necesario (ej. permitir una buena distribución de un agente químico después de ser añadido al agua).

Tabla 12. Condiciones de entrada y niveles de salida para cada uno de los estados del proceso de control de depuración

PROCESO: DEPURACIÓN						
Estado	Input/Output	Factor A	Con	Factor B	Op	Descripción
READY	INPUT	Always				La condición de entrada al estado READY siempre se cumple.
	OUTPUT	Depuración	.=	OFF		Depuradora inactiva
START	INPUT	<i>TimeOut</i>	.=	3 horas		El proceso de depuración se detendrá 3 hora después de su inicio
		Fecha y Hora	.=	21:00	AND	El proceso de depurado comienza a las 21:00
		Fecha y Hora	<	1-Junio	AND	Desde el 1 de Junio
		Fecha y Hora	>	30-Sept	OR	Hasta el 30 de Septiembre
		Terminal: GENERAL - Boton Izq.	.=	ON		El proceso puede activarse manualmente desde el terminal de visualización GENERAL, pulsando el botón izquierdo.
	OUTPUT	Depuración	.=	ON		Depuradora en funcionamiento
STOP	INPUT					No existen condiciones de entrada, al proceso anterior se la ha asignado un <i>TimeOut</i>
	OUTPUT	Depuración	.=	ON		Depuradora en funcionamiento
BLOCKED	INPUT					
	OUTPUT					

e) Alarma

Detección de presencia (proceso ALARMA): Se ha instalado un sistema de alarma con el fin de detectar la entrada de intrusos en el interior de la parcela, en especial, en el perímetro de la casa. Para ello se han conectado al sistema de monitorización 2 sensores tipo PIR (Passive Infrared Sensor), S5 y S6 a través de los cuales el sistema detecta la presencia de intrusos. Si los sensores detectan presencia, activarán un avisador acústico mediante el actuador A5. A su vez, la unidad coordinadora enviará un mensaje por correo electrónico a una dirección determinada. Tanto los 2 elementos de detección como el actuador se encuentran conectados al nodo NF4.

La alarma se detendrá cuando el nodo NF5 (configurado como nodo de localización de personal) se encuentre en el área de cobertura de la unidad coordinadora.

Tabla 13. Condiciones de entrada y niveles de salida para cada uno de los estados del proceso de control alarma

PROCESO: ALARMA						
Estado	Input/Output	Factor A	Con	Factor B	Op	Descripción
READY	INPUT	Always				La condición de entrada al estado READY siempre se cumple.
	OUTPUT	Alarma	.=	OFF		Zumbador de la alarma inactivo
START	INPUT	<i>Send e-mail notification when input condition</i>				Envía un mensaje por correo electrónico indicando que el proceso se ha activado
		<i>TimeOut</i>	.=	12 horas		La alarma dejará de sonar 12 horas después de su activación
		Presencia Izquierda	.=	ON	OR	El proceso se activará cuando se detecte presencia en el sensor izquierdo
		Presencia Derecha	>	ON		o en el sensor derecho
	OUTPUT	Alarma	.=	ON		Alarma en funcionamiento
STOP	INPUT	Presencia	.=	Localizador	OR	Si el nodo “Localizador” se encuentra dentro del área de cobertura, la alarma dejará de sonar
		Terminal: GENERAL - Boton Derecho	.=	ON		La alarma también puede ser desactivada remotamente a través del botón derecho del terminal de visualización GENERAL
	OUTPUT	Alarma	.=	OFF		Alarma desactivada
BLOCKED	INPUT	Presencia	.=	Localizador		Si el nodo “Localizador” se encuentra dentro del área de cobertura, el proceso queda bloqueado y la alarma nunca se activará
	OUTPUT	Alarma	.=	OFF		Alarma desactivada

II.6.5. Terminal de visualización

Todos los parámetros relevantes recogidos por el sistema en el escenario bajo monitorización, pueden ser visualizados en una única pantalla de terminal de visualización.

Para ello, se pone como ejemplo la siguiente distribución de elementos:

Tabla 14. Ejemplo de terminal de visualización adaptado a la aplicación de control para finca rústica

GENERAL	
DEPURACIÓN (P)	CONTROL DEPÓSITO (P)
	ALARMA (P)
RIEGO CÉSPED (P)	Humedad Césped (AI)
RIEGO FRUTALES (P)	Humedad Frutales (AI)
Presencia Izquierda (DI)	Llenado Superior (DI)
Presencia Derecha (DI)	Llenado Inferior (DI)

Los procesos (P) de la barra superior (3 en línea) serán mostrados con su estado (STOP, RUNNING, etc). Los procesos de la tabla de datos central mostrarán su estado mediante un indicador que simboliza los distintos estados del proceso con distintos colores (ver sección “II.5.1.h. *Visualization Terminals*”).

Las entradas digitales (DI) también serán indicadas con colores distintos según el estado, mientras que las entradas analógicas (AI) mostrarán el valor leído convertido a unidades físicas, en este caso Humedad Relativa.

Con el fin de disminuir el tráfico de datos entre el terminal de visualización y la unidad coordinadora, y teniendo en cuenta la frecuencia de variación de los datos mostrados, se propone una configuración para el parámetro *data refresh rate* de 5 segundos.

II.7. Resultados y Conclusiones

El sistema desarrollado permite la adaptación de la tecnología WSN tanto al escenario real propuesto como ejemplo de aplicación, como a gran número de aplicaciones gracias a la flexibilidad y la filosofía de funcionamiento, totalmente independiente del HW y con solo 3 elementos básicos diferenciados a nivel de usuario: nodos inalámbricos, GPDAIOs y terminales de visualización.

El interfaz de configuración, basado en Web y accesible desde cualquier explorador, permite al usuario configurar el sistema y realizar modificaciones sobre este de manera sencilla y sin la necesidad de conocimientos técnicos.

No obstante, existen limitaciones en el mismo, por ejemplo, en cuanto tiempos de ejecución de ordenes, ya que la filosofía intrínseca de las redes WSN basadas en el estándar IEEE 802.15.4 es no determinista en cuanto a latencias se refiere, existiendo retardos de pocos segundos en el peor de los casos.

A continuación se citan de manera resumida algunas características a destacar en el sistema desarrollado:

- Sencillez de manejo y configuración.
- No requiere conocimientos técnicos (totalmente manejable a nivel de usuario).
- El número de nodos inalámbricos permitidos en el sistema alto (la red ZigBee Pro soporta hasta 64000).
- El código fuente permite la integración y coexistencia de distintas tecnologías inalámbrica en un futuro (ver sección "II.8. Trabajo Futuro").
- Capacidad de lectura de entradas tipo analógico y digital.
- Capacidad de actuación con salidas tipo analógico y digital.
- Permite el uso de dispositivos alimentados con baterías (bajo consumo).
- Accesibilidad remota tanto en la visualización como en la configuración.
- Gran flexibilidad en cuanto a la configuración de lógica de procesos.
- Se permiten enclavamientos a varios niveles.
- Permite la ejecución de órdenes remotas a través de los terminales de visualización.
- Disparo de eventos temporales (fecha y hora o *TimeOuts*).
- Disparo de eventos derivados de la presencia, o no, de nodos en la red.
- Bajo coste.
- Instalación sencilla.

II.8. Trabajo Futuro

Con el fin de mejorar el sistema de cara a una posible comercialización, se proponen las siguientes tareas como trabajos a realizar en el futuro:

- Encriptación datos:

Actualmente todos los datos intercambiados entre la unidad principal y los terminales de visualización y ordenador desde el que se realiza la configuración se transmiten usando un *socket* TCP estándar sin encriptación de datos.

Esto permite a un posible *sniffer* conectado a la red, capturar los datos y realizar actuaciones sobre el sistema o cambios en la configuración sin la intervención del usuario.

La implementación de protocolos de intercambio seguros, tipo HTTPS, permitirá la encriptación de los datos transmitidos, haciendo prácticamente imposible su inteligibilidad en caso de ser capturados por un tercero.

- Discriminación de presencia en distintas zonas:

Actualmente se puede establecer como parámetro de entrada al estado de un proceso, la presencia de un nodo inalámbrico dentro de la red. Con un desarrollo adecuado, el sistema sería capaz de discriminar si el nodo transmisor de mensajes de localización (*tracks*) se encuentra cerca de uno u otro nodo tipo "R", permitiendo discriminar entre distintas zonas dentro de la red. Esto permitiría por ejemplo, indicar presencia dentro de distintos recintos en áreas de gran cobertura (hay que tener en cuenta que una misma red puede albergar cientos o miles de nodos).

- Registro de datos y gráficas:

En determinadas aplicaciones o durante procesos de diagnóstico, resulta útil disponer de la evolución de los datos de una determinada señal a lo largo de un determinado periodo de tiempo (ejemplo, la evolución de la temperatura durante el invierno en una plantación para determinar si es necesario la instalación de un sistema anti-heladas).

Con el correspondiente desarrollo, el sistema sería capaz de almacenar los datos en la tarjeta de memoria SD en formato, por ejemplo, .CVS, de forma que pudieran extraerse o descargarse a través de la red hacia un tercer dispositivo para su análisis posterior.

La posibilidad de visualizar los gráficos vía Web también sería viable con un desarrollo adicional.

- Mejorar eficiencia de tráfico de datos:

Es posible mejorar la eficiencia de tráfico de datos entre la unidad coordinadora y los terminales de visualización. Actualmente, es el terminal de visualización el que pregunta a la unidad coordinadora por los datos, y ésta los envía tanto si están o no actualizados.

Con un futuro desarrollo, podría ser la unidad coordinadora la que envíe datos al terminal de visualización solo en caso de que estos hayan cambiado desde el último envío realizado.

Otra posible técnica a implementar es la pregunta por parte del terminal a la unidad coordinadora por nuevos datos, siendo esta la que determine si han sido actualizados y en este caso, enviar únicamente los que han cambiado.

- Botones con clave de seguridad:

Actualmente los botones disponibles en los terminales de visualización son capaces de actuar sobre procesos, algunos de ellos de seguridad, como por ejemplo, la activación o desactivación de una alarma. En estos casos, al pulsar el botón, el terminal deberá pedir una clave de identificación al usuario de forma que no permita la manipulación del proceso por un tercero.

- Ampliación a otras tecnologías *wireless*:

Actualmente el sistema implementa un interfaz radio ZigBee que proporciona comunicación inalámbrica y gestiona la red. No obstante, para aplicaciones sencillas o cuando se requiere un gran alcance directo (sin la existencia de repetidores intermedios), hay otras tecnologías que son

mas apropiadas.

El uso de sistemas propietarios (no ZigBee, ni IEEE 802.15.4) en 2.4Ghz permitirían una reducción de costes importantes, sobretodo proporcionalmente en los nodos remotos, y que sacrificando ciertas capacidades (ej. saltos y enrutado de información), pueden satisfacer multitud de aplicaciones domésticas.

La transmisión radio en otras frecuencias mas bajas de la banda ISM (868 y 433 Mhz) permiten obtener un mayor alcance utilizando la misma potencia radio, sacrificando ancho de banda, que a su vez, se ve compensado por la simplificación de la pila de protocolos.

El código fuente dependiente del interfaz radio se ha preparado para la incorporación de otros dispositivos (módulos radio) de forma sencilla y garantizando su coexistencia. A continuación se muestra un fragmento de código en el que se puede ver un ejemplo de la estructura con reserva para añadir otros tipos de nodos inalámbricos en un futuro.

```
// Lectura del nombre del nodo
pParam = HTTPGetROMArg(curHTTP.data, (ROM_BYTE *) "nname");
if(pParam)
{
    if(*(WSN_NODE_TYPE*)WSN.node[nodeIndex]==ETRX2_NODETYPE)
    {
        strncpy(((ETRX2_NODE*)WSN.node[nodeIndex])->name, (const char*)pParam,
                sizeof(ETRX2[0].name));
    }
    // if: ESPACIO RESERVADO PARA INCORPORAR OTROS TIPOS DE NODOS INALÁMBRICOS
}
```

Actualmente se están realizando desarrollos en este sentido, aunque no estarán finalizados en la entrega de este TFG.

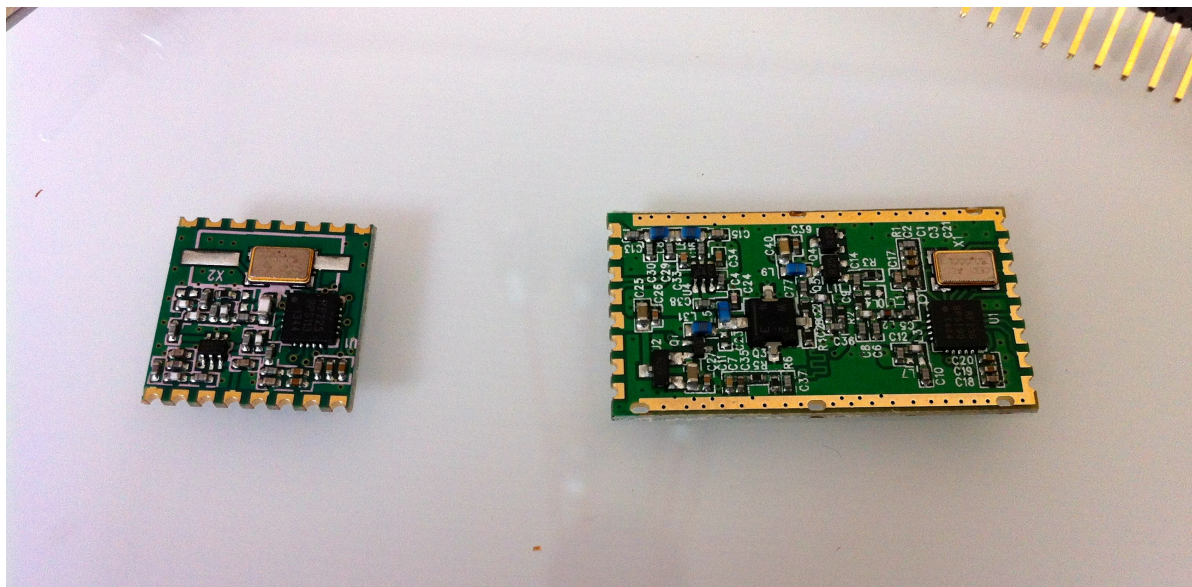


Figura 43. Módulos radio transceptores de 100mW (izquierda) y 1W (derecha) de potencia a 433 Mhz basado en el integrado SI4432 (Silicon Labs) para ser añadido como segundo interfaz radio.

- Ampliación de memoria RAM y Flash en el microcontrolador: Actualmente el microcontrolador no es capaz de almacenar la información correspondiente a los 64K nodos admisibles en una red tipo ZigBee Pro debido a que el espacio de memoria RAM está limitado a 128KB. La incorporación de

una memoria RAM externa permitiría incrementar el número de todos, procesos y GPDAIOs máximo del sistema.

De igual modo, los 512KB de memoria Flash limitan la cantidad de líneas de código, y sobretodo de código HTML, que el servidor Web es capaz de procesar y transferir. Con una ampliación de la memoria Flash se podrían conseguir páginas Web mas elaboradas con un entorno gráfico mejorado basado en HTML5 y con, por ejemplo, chequeo de parámetros en Web antes de realizar el envío al servidor (ejemplo, implementación del API jQuery).

- *Bootloader*: Con el fin de facilitar las actualizaciones de FW del microcontrolador de la unidad coordinadora, se propone como trabajo futuro la implementación de un *Bootloader* que permita leer una imagen guardada en la tarjeta SD y actualizar el programa fuente de la flash con la nueva versión.

III. DIAGRAMAS Y PLANOS

III.1.Unidad Coordinadora

III.1.1.Tarjeta PIC32 Ethernet Starter Kit

1.2 PIC32 FUNCTIONALITY AND FEATURES

Representations of the layout of the PIC32 Ethernet Starter Kit are shown in [Figure 1-1](#) and [Figure 1-2](#).

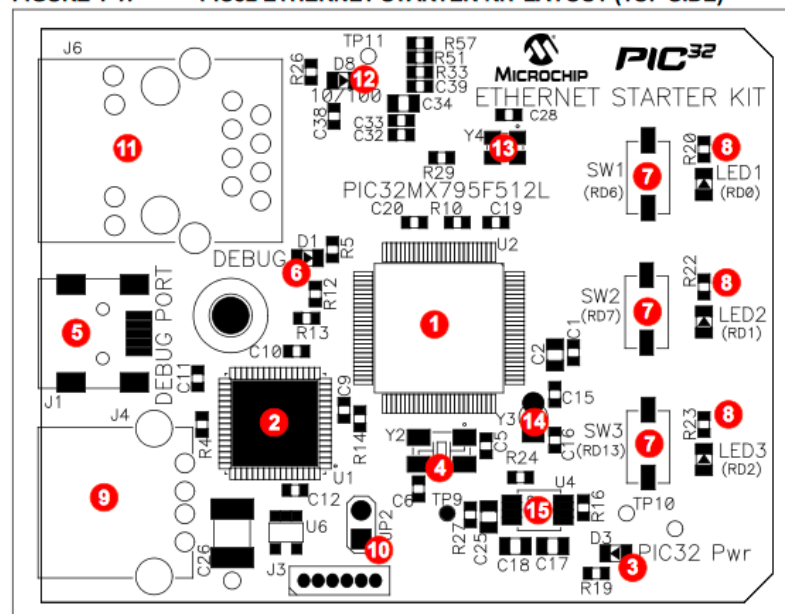
The top assembly of the board includes these key features, as indicated in [Figure 1-1](#):

1. PIC32MX795F512L 32-bit microcontroller.
2. PIC32MX440F512H USB microcontroller for on-board debugging.
3. Green power indicator LED.
4. On-board crystal for precision microcontroller clocking (8 MHz).
5. USB connectivity for on-board debugger communications.
6. Orange debug indicator LED.
7. Three push button switches for user-defined inputs.
8. Three user-defined indicator LEDs.
9. USB Type A receptacle connectivity for PIC32 host-based applications.
10. HOST mode power jumper.
11. RJ-45 Ethernet port.
12. Ethernet 10/100 bus speed indicator LED.
13. 50 MHz Ethernet PHY oscillator.
14. 32 kHz oscillator (optional).
15. USB Host and OTG power supply for powering PIC32 USB applications.

Note: When running self-powered USB device applications, open the jumper JP2 to prevent possibly back-feeding voltage onto the VBUS from one port on the host to another (or from one host to another).

For details on these features, refer to [Chapter 2. "Hardware"](#).

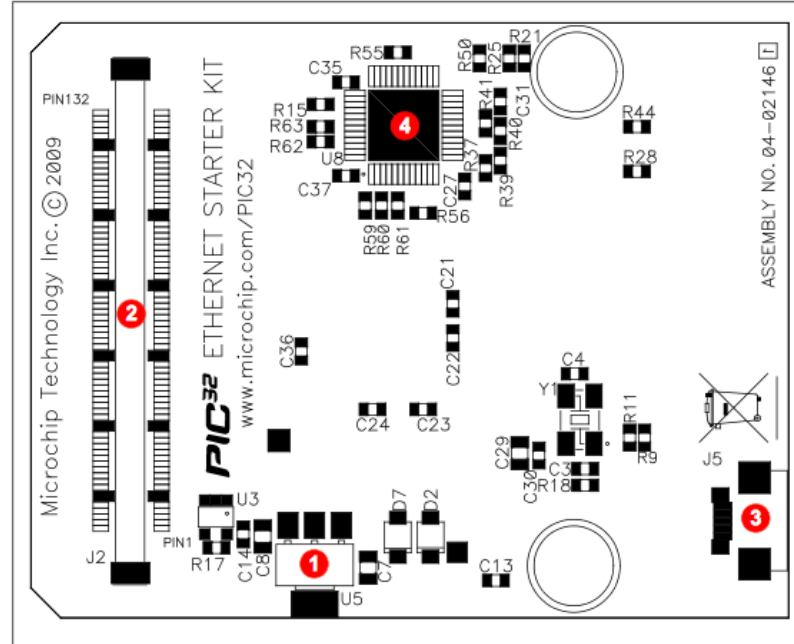
FIGURE 1-1: PIC32 ETHERNET STARTER KIT LAYOUT (TOP SIDE)



The bottom assembly of the board includes these key features, as indicated in Figure 1-2:

1. Regulated +3.3V power supply for powering the starter kit via USB or expansion board.
2. Connector for various expansion boards.
3. USB Type micro-AB receptacle for OTG and USB device connectivity for PIC32 OTG/device-based applications.
4. External Ethernet PHY.

FIGURE 1-2: PIC32 ETHERNET STARTER KIT LAYOUT (UNDERSIDE)



Appendix A. Board Layout and Schematics

A.1 PIC32 ETHERNET STARTER KIT BLOCK DIAGRAM

FIGURE A-1: HIGH-LEVEL BLOCK DIAGRAM OF THE PIC32 ETHERNET STARTER KIT

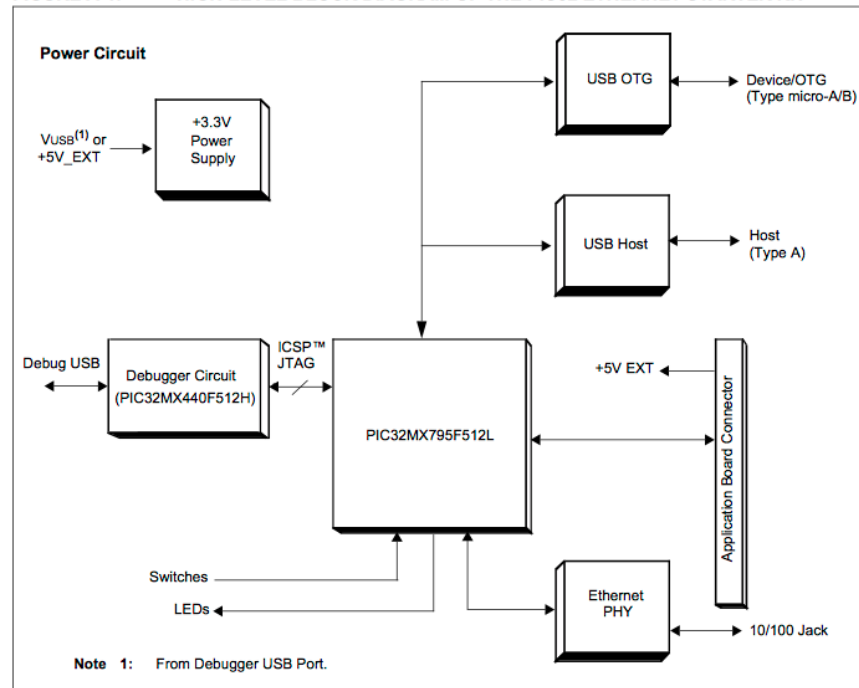


FIGURE A-2: PIC32 ETHERNET STARTER KIT LAYOUT (TOP ASSEMBLY)

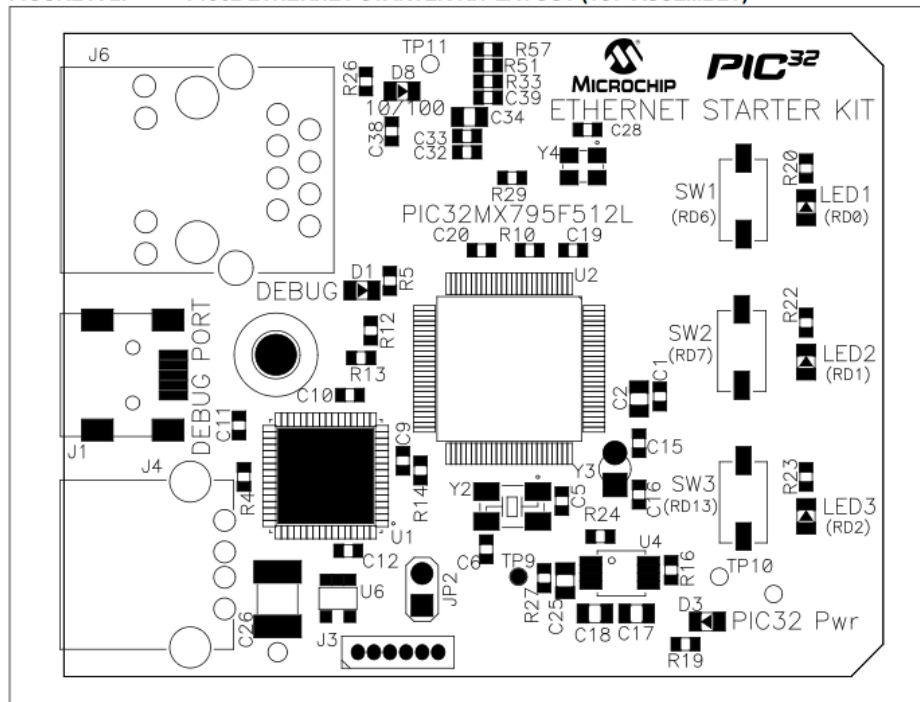
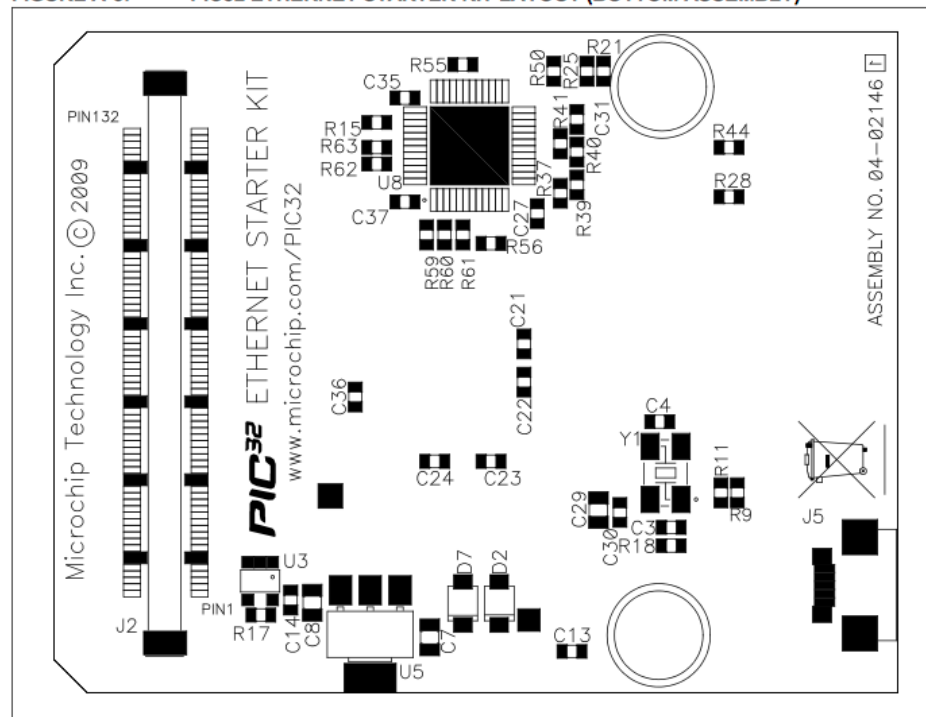


FIGURE A-3: PIC32 ETHERNET STARTER KIT LAYOUT (BOTTOM ASSEMBLY)



A.3 PIC32 ETHERNET STARTER KIT BOARD SCHEMATICS

FIGURE A-4: PIC32 ETHERNET STARTER KIT (ETHERNET)

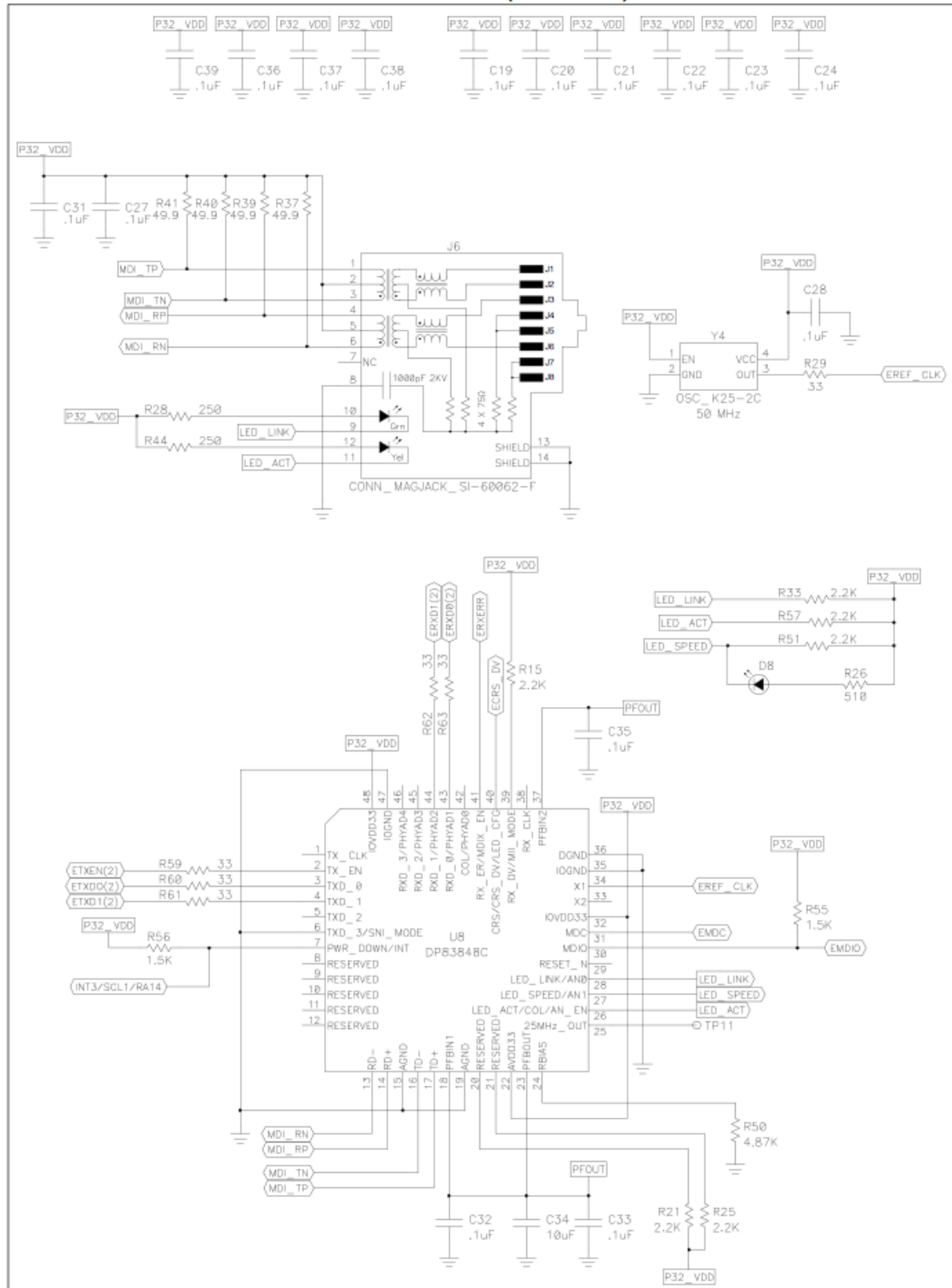


FIGURE A-5: PIC32 ETHERNET STARTER KIT (PIC32 DEVICE)

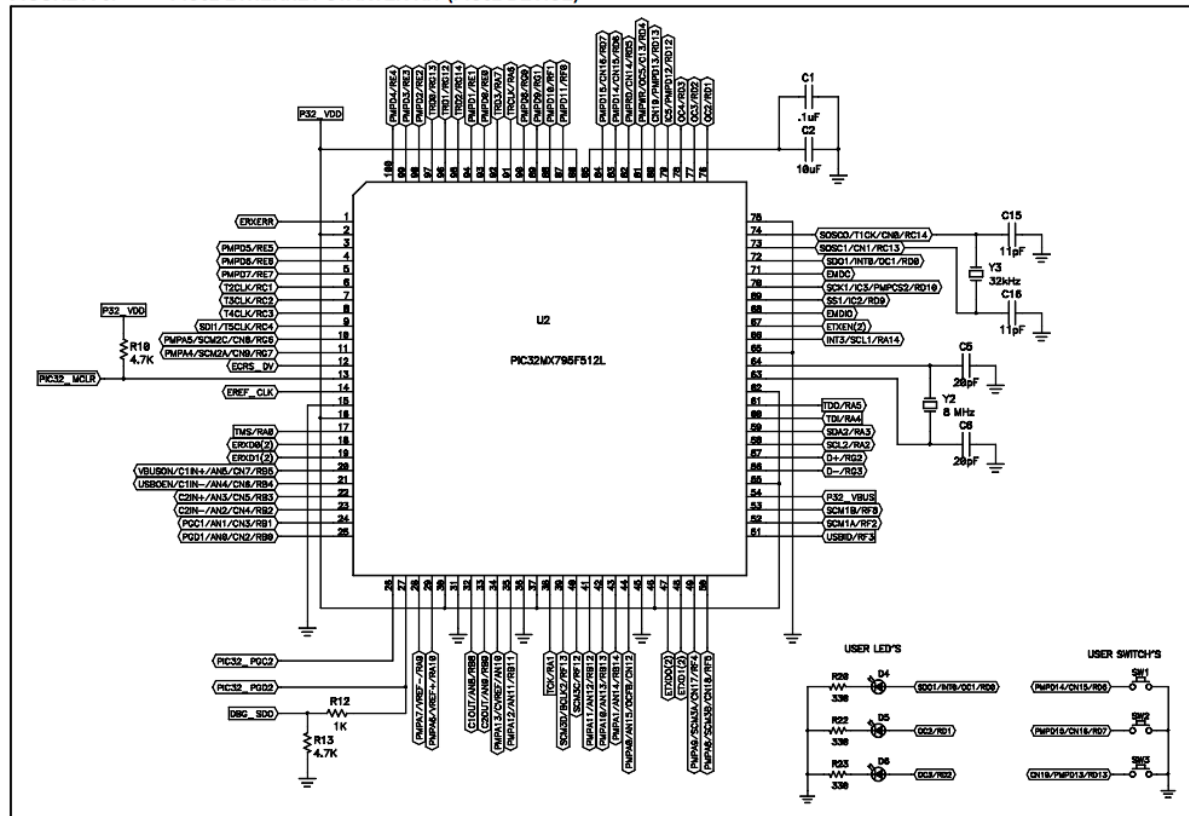


FIGURE A-6: PIC32 ETHERNET STARTER KIT (USB HOST AND OTG POWER SUPPLIES)

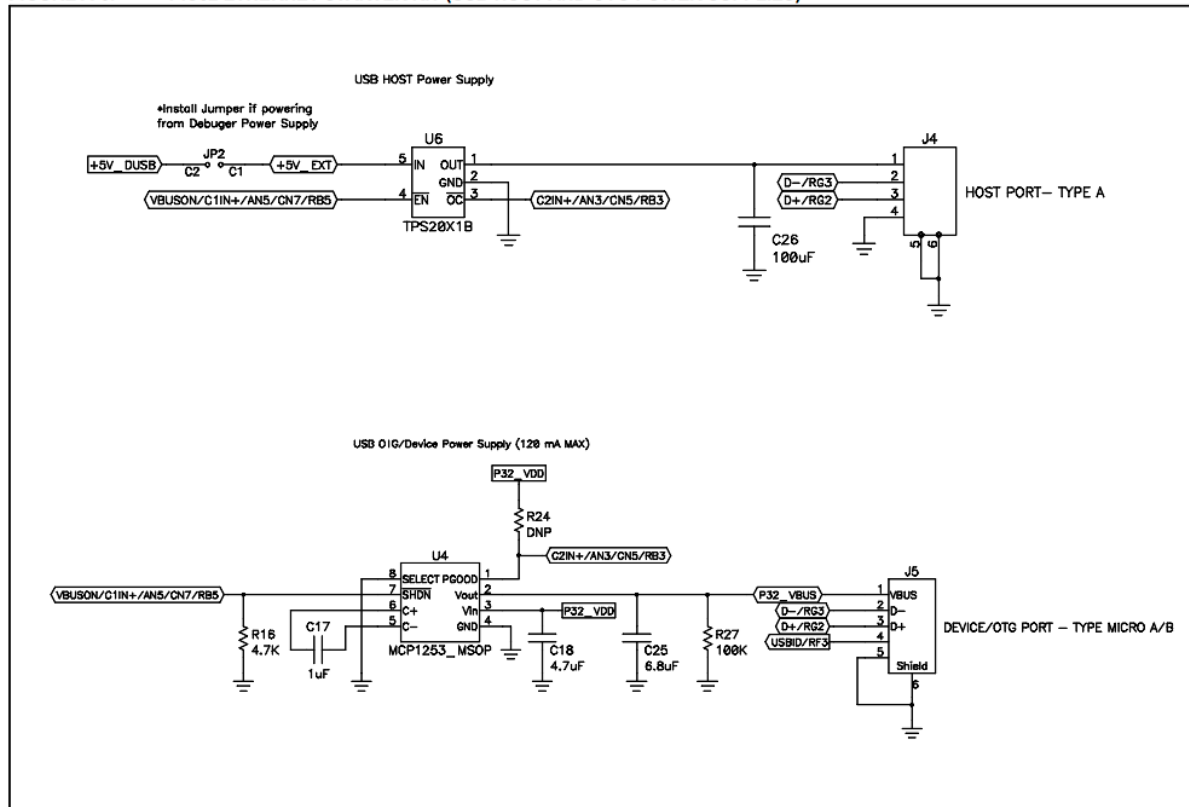


FIGURE A-7: PIC32 ETHERNET STARTER KIT (DEBUGGER)

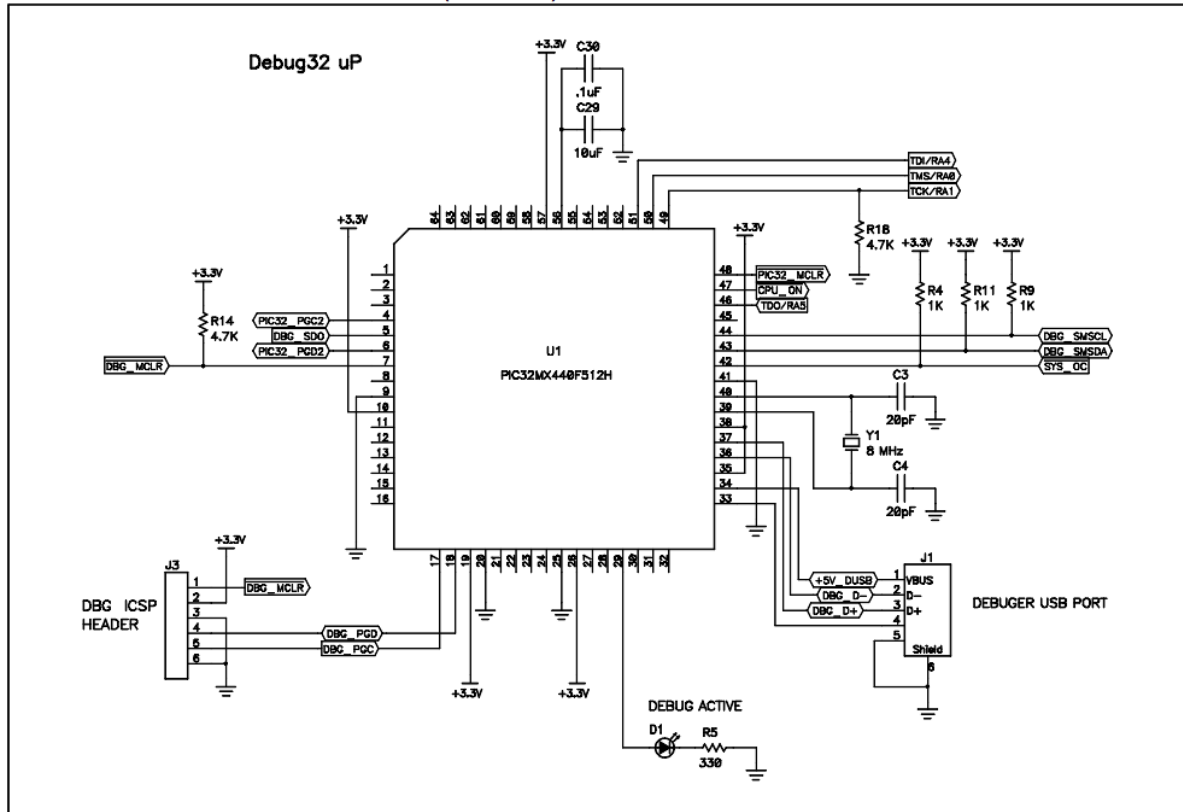


FIGURE A-8: PIC32 ETHERNET STARTER KIT (3.3V POWER SUPPLY)

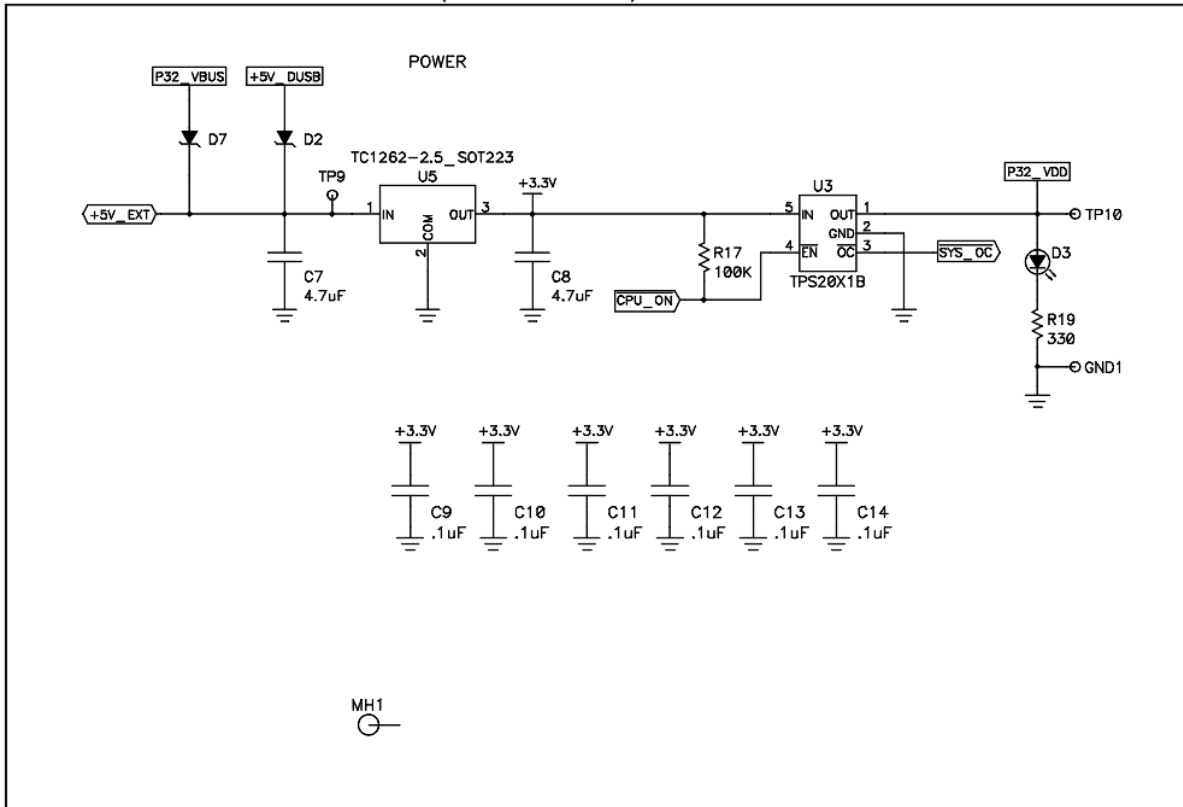
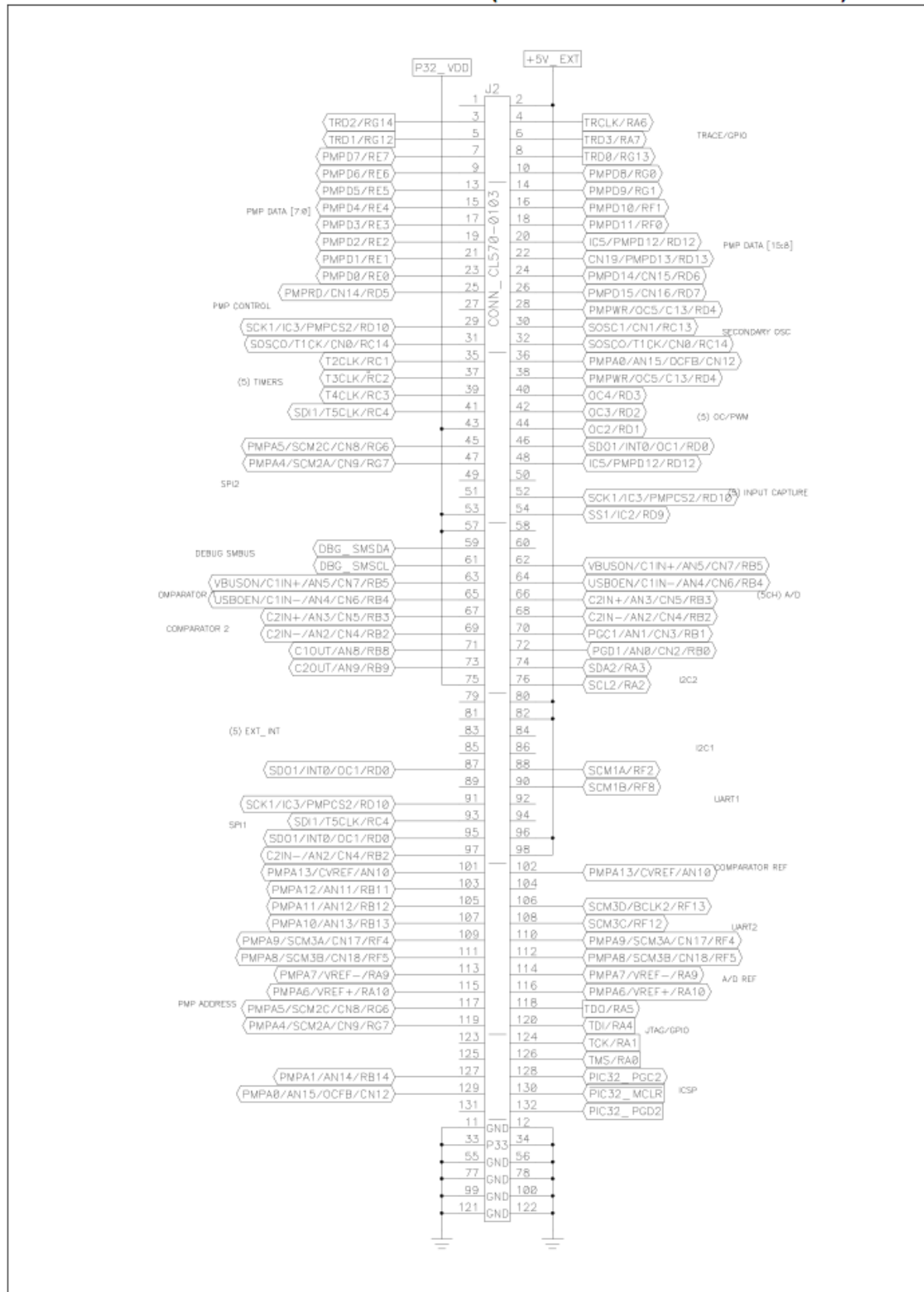
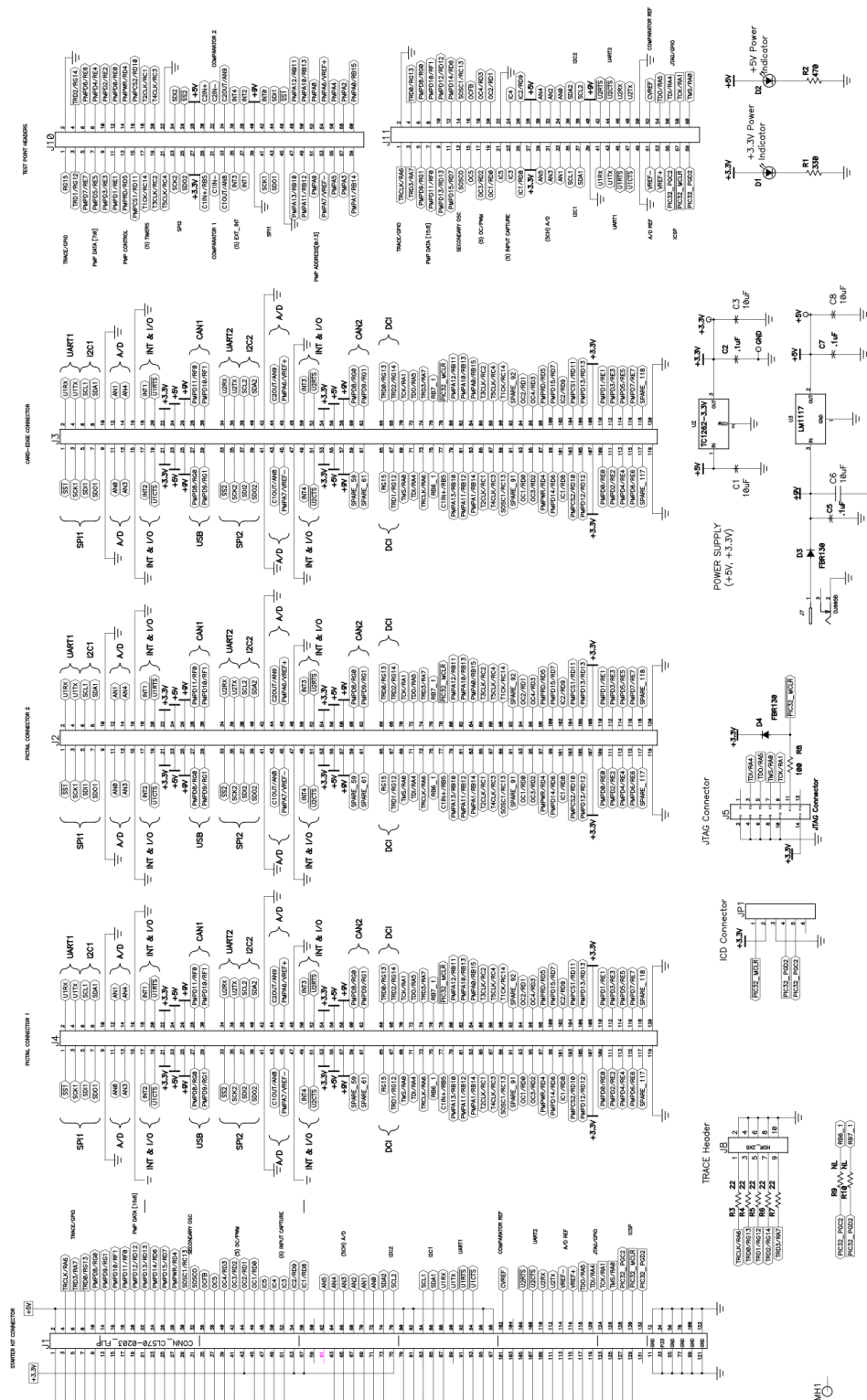
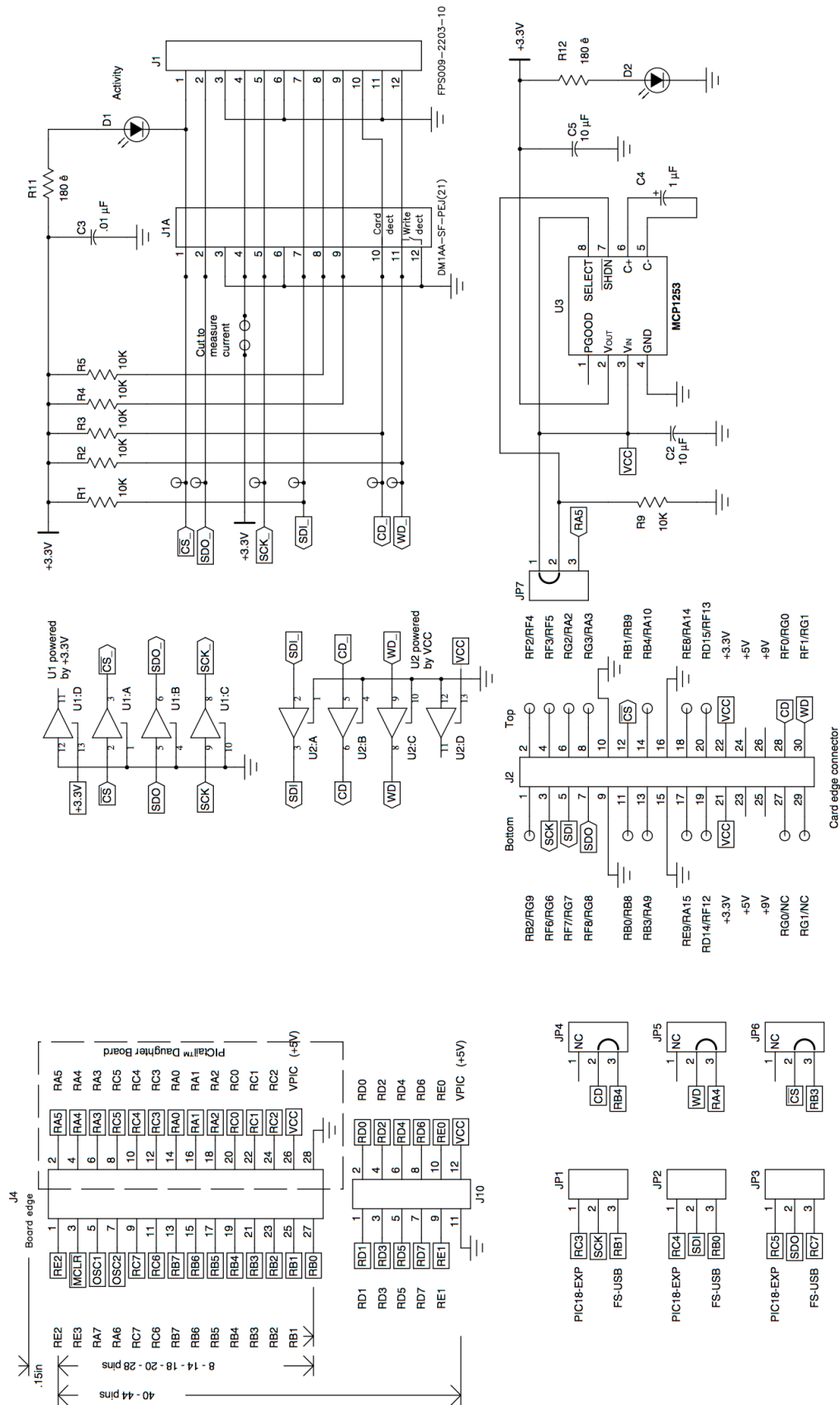


FIGURE A-9: PIC32 ETHERNET STARTER KIT (APPLICATION BOARD CONNECTOR)


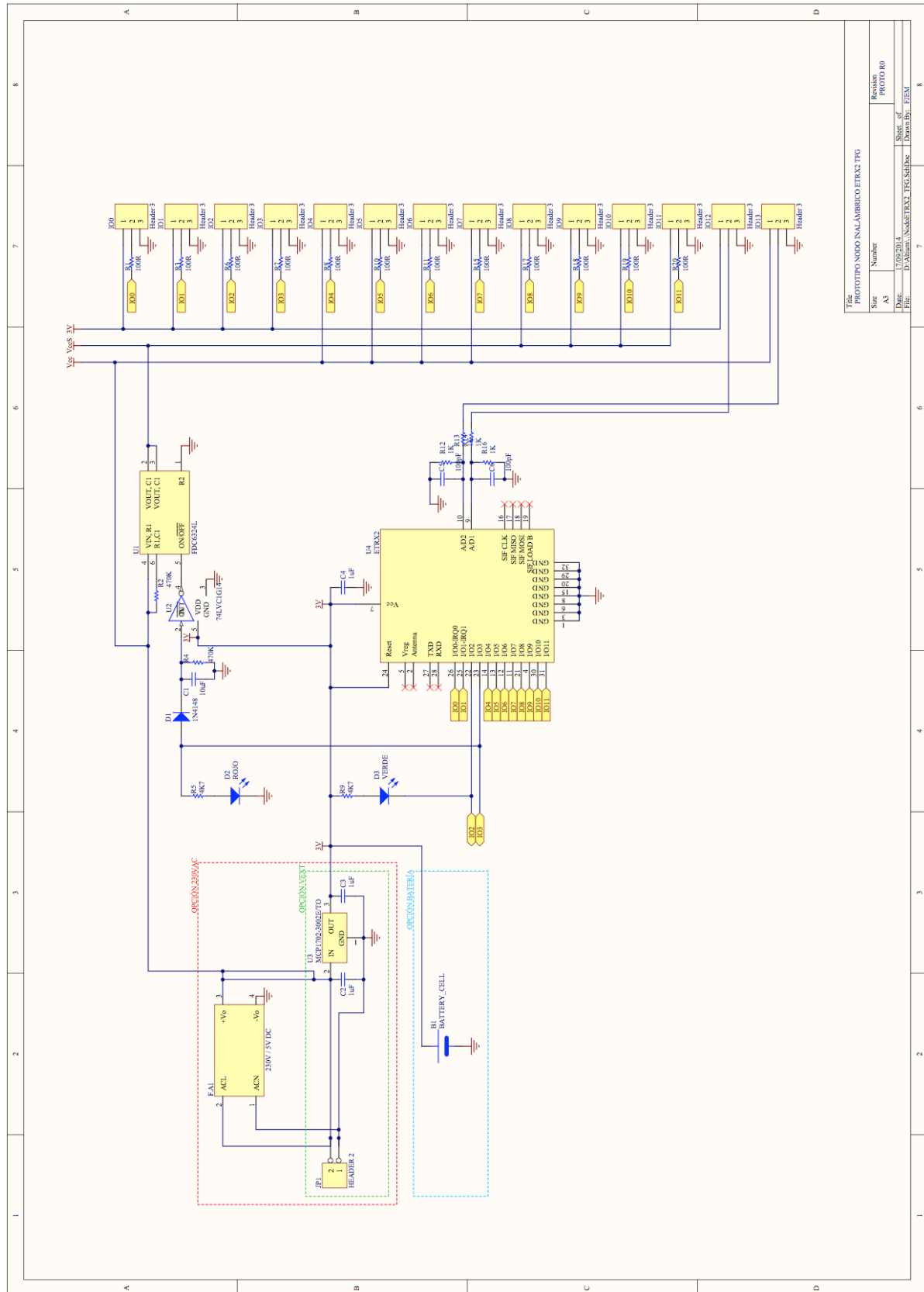
III.1.2. Tarjeta Starter Kit IO Expansion Board



III.1.3.Tarjeta PICtail Daughter Board for SD and MMC Cards



III.2. Prototipo de nodo inalámbrico



IV. SOFTWARE

IV.1. Principales Estructuras de los Objetos de Red

El siguiente extracto muestra los objetos remotos principales de una red WSN (GPDAIOs y nodos). Tomando como referencia la estructura “ETRX2_NODE”, se pueden integrar en el sistema otros tipos de nodos inalámbrico (distinto HW).

```

/*
 * File:   WSNOBJETS.h
 * Author: Fran
 *
 * Created on March 21, 2014, 12:28 PM
 */

// Definiciones de tipos de objetos que se pueden encontrar en una red WSN

#ifndef WSNOBJETS_H
#define WSNOBJETS_H

#include "GenericTypeDefs.h"

// GPDAIOs
*****
*****
// Tipos de entradas posibles
typedef enum GPDAIO_TYPE{
    UNDEFINED_TYPE = 0b00000000,
    DIGITAL_INPUT = 0b00000001,
    DIGITAL_OUTPUT = 0b00000010,
    ANALOG_INPUT = 0b00000100,
    PWM_OUTPUT = 0b00001000
}GPDAIO_TYPE;

// Estados logicos para entradas/salidas digitales
#define IO_ON 1.00
#define IO_OFF 0.00

// DEFINICION GPDAIO (Definicion general para una General Purpose Digital and Analog Input/
// Output)
typedef struct GPDAIO_OBJECT{
    void *node; // Puntero al nodo al que pertenece esta IO
    void **nodeGPDAIO; // Puntero recursivo al puntero que apunta a esta entrada (la parte
                        // contrarante de la primera parte...). Se pone a cero cuando se
                        // elimina la entrada y sirve para indicar que el puerto de ese
                        // nodo esta libre.

    float val; // Valor de la entrada (tanto si es entrada como si es salida) (para una
                // PWM estara expresado en porcentaje)
    float oldVal; // Valor de la entrada en el pase anterior (permite detectar cambios)

    // Nombres
    char name[20]; // Nombre de la entrada/salida (maximo 19 caracteres)
    char units[8]; // Nombre de las unidades asignadas a la entrada/salida (maximo 7
                  // caracteres)

    // Escalados y conversiones a unidades fisicas
    float signalMin; // Valor minimo a la entrada/salida
    float signalMax; // Valor maximo a la entrada/salida
    float unitMin; // Valor minimo de la unidad para el valor valMin
    float unitMax; // Valor maximo de la unidad para el valor valMax

    // Tipo de entrada
    GPDAIO_TYPE type __attribute__((packed));

    UINT16 newValueUpdatePending:1; // Hay pendiente procesar un cambio producido en
                                    // valor de entrada.
    UINT16 digitalInPolarityInverse:1; // Indica que la polaridad de la entrada digital
                                    // esta invertida (High = OFF)
    UINT16 digitalOutPolarityInverse:1; // Indica que la polaridad de la salida digital
                                    // esta invertida (High = OFF)
}GPDAIO_OBJECT;

// Macro para obtener el indice de una GPDAIO en el array de GPDAIOs
#define getGPDAIOindex(pGPDAIO) (unsigned int)(pGPDAIO-GPDAIO)

```



```

// NODOS
// Tipos de nodos
*****
typedef enum{
    EMPTY_NODETYPE = 0,
    ETRX2_NODETYPE
}WSN_NODE_TYPE;

// DEFINICIONES PARA ETRX2
*****
#define ETRX2_MAX_GPDAIO    14        // Numero total de entradas y salidas para este tipo de
                                        nodo.

typedef struct ETRX2_NODE{
    WSN_NODE_TYPE type;                // Tipo de nodo
    UINT64_VAL MAC;                    // MAC ADDRESS del nodo en particular: 64-bit IEEE 802.15.4
    address in hexadecimal
    UINT64_VAL parentMAC;              // MAC ADDRESS del nodo padre a este nodo: 64-bit IEEE
                                        802.15.4 address in hexadecimal
    UINT64_VAL sinkMAC;                // MAC ADDRESS del nodo sink que recogerá los datos: 64-bit
                                        IEEE 802.15.4 address in hexadecimal

    char name[18];                     // Nombre del nodo (maximo 16 caracteres)

    char linkQuality;                  // Calidad del enlace con el nodo
    char RSSI;                          // Nivel de senial RSSI con el nodo

    UINT16 AD1;                        // Lectura de la entrada analogica 1 en mV
    UINT16 AD2;                        // Lectura de la entrada analogica 2 en mV

    UINT16 IODir;                      // Direcciones de los puertos IO (1=salida, 0=entrada)
    UINT16 IOinput;                    // Estado de los pines IO de entrada (1=nivel alto, 0=nivel bajo)
    UINT16 IOoutput;                   // Estado de los pines IO de salida (1=nivel alto, 0=nivel bajo)
    UINT16 IOpullUP;                   // Habilita las resistencias de pullUP internas (1=habilitado, 0=no
                                        habilitado) (necesita reseteo)
    UINT16 IOpulDOWN;                  // Habilita las resistencias de pullDOWN internas (1=habilitado,
                                        0=no habilitado) (necesita reseteo)

    UINT16 PWMfreq;                    // Frecuencia de la senial PWM fijada segun: Frequency = 12MHz/(
                                        {PWMfreq}+1)
    UINT16 PWMdutyCycle;                // Ancho de impulso de la senial PWM segun lo siguiente: Duty
                                        cycle = {PWMdutyCycle}/{PWMfreq}+1

    // Entradas y salidas (el total) 0-11 GPIO, 12-13 Analog Inputs
    GPDAIO_OBJECT *GPDAIO[ETRX2_MAX_GPDAIO];

    //Flags de configuracion y estado
    enum __attribute__((packed)){
        FFD,
        MED,
        SED
    }nodeType;
    unsigned PWMpinEnabled:1;           // Define IO3 como pin PWM

    unsigned confiUpdatePending:1;     // Hay pendiente una actualizacion de los datos de
                                        configuracion del dispositivo
} ETRX2_NODE;

// Definicion de tipos de entrada para ETRX2
static const GPDAIO_TYPE ETRX2_GPDAIO_TYPE_DEFINITION[] = {
    DIGITAL_INPUT | DIGITAL_OUTPUT,    //IO0
    DIGITAL_INPUT | DIGITAL_OUTPUT,    //IO1
    DIGITAL_INPUT | DIGITAL_OUTPUT,    //IO2
    DIGITAL_INPUT | DIGITAL_OUTPUT | PWM_OUTPUT, //IO3
    DIGITAL_INPUT | DIGITAL_OUTPUT,    //IO4
    DIGITAL_INPUT | DIGITAL_OUTPUT,    //IO5
    DIGITAL_INPUT | DIGITAL_OUTPUT,    //IO6
    DIGITAL_INPUT | DIGITAL_OUTPUT,    //IO7
    DIGITAL_INPUT | DIGITAL_OUTPUT,    //IO8
    DIGITAL_INPUT | DIGITAL_OUTPUT,    //IO9
    DIGITAL_INPUT | DIGITAL_OUTPUT,    //IO10
    DIGITAL_INPUT | DIGITAL_OUTPUT,    //IO11
    ANALOG_INPUT,                      //AD1
    ANALOG_INPUT                        //AD2

```

```
};

#endif /* WSNOBJETS_H */
```

IV.2. Gestión de nodos inalámbricos genéricos

Gestión de objetos de red. La estructura de red “WSN_PARAMETERS” contiene punteros genéricos (tipo *void**) a los nodos inalámbricos. Es la propia estructura del nodo la que informa sobre su tipo, según el *enum* “WSN_NODE_TYPE”.

```
/*
 * File:   WSN.h
 * Author: Fran
 *
 * Created on March 21, 2014, 2:20 PM
 */

#ifndef WSN_H
#define WSN_H

#include "WSNobjects.h"

#ifdef USE_SD
    #include "SDcard.h"
    #define SaveWSNConfig()      saveDataFile("ETRX2.dat", (void*)&ETRX2, sizeof(ETRX2));\
                                saveDataFile("WSN.dat", (void*)&WSN, sizeof(WSN));\
                                saveDataFile("GPDAIO.dat", (void*)&GPDAIO,\
                                sizeof(GPDAIO));

    #define ReadWSNConfig()     readDataFile("ETRX2.dat", (void*)&ETRX2, sizeof(ETRX2));\
                                readDataFile("WSN.dat", (void*)&WSN, sizeof(WSN));\
                                readDataFile("GPDAIO.dat", (void*)&GPDAIO,\
                                sizeof(GPDAIO));
#endif

// Definiciones generales
#define MAX_SYSTEM_NODES      32
    #define MAX_ETRX2_NODES    32

#define MAX_SYSTEM_GPDAIO     128

// Macros
// Devuelve el puntero tipo del nodo correspondiente al indice dado como NO NULL.
#define getNodePointer(index)\
    ETRX2_NODE *localETRX2 = NULL;\
    switch(WSN.nodeType[index]){\
        case ETRX2_NODETYPE: localETRX2 = &WSU.node[index]; break;\
    }

// Estructura global de la red WSN
typedef struct WSN_PARAMETERS{
    WORD nodeNumber;          // Numero total de nodos existentes en la red WSN
    void *node[MAX_SYSTEM_NODES]; // Puntero a los nodos existentes
    WORD GPDAIONumber;        // Numero total de GPDAIOs configuradas en el sistema
}WSN_PARAMETERS;

// Datos
//extern ETRX2_NODE ETRX2[MAX_ETRX2_NODES];
extern ETRX2_NODE ETRX2[MAX_ETRX2_NODES];

extern GPDAIO_OBJECT GPDAIO[MAX_SYSTEM_GPDAIO];
extern WSN_PARAMETERS WSN;

//Funciones
extern int initWSN(void);
extern int addWSNnode(void* nodeID, WSN_NODE_TYPE nodeType);
extern int delWSNnode(UINT16 nodeIndex);
```

```
extern int addGPDAIO(void* pNodo);

#endif /* WSN_H */
```

IV.3. Generación de código HTML dinámico en servidor Web

El servidor Web detecta variables de código dinámico (identificadas en HTML como ~variable~) y ejecuta una llamada a la función HTTPPrint_~variable~. Dichas funciones, como la del ejemplo de abajo, se encargan de crear el contenido HTML correspondiente a la variable (que a veces puede ser de cientos de caracteres), verificar si el *buffer* de transmisión disponible es suficiente y enviar el código correspondiente, en función del estado del sistema, mediante la función "TCPPut".

En caso de no existir suficiente memoria en el *buffer* de salida como para enviar toda la información necesaria, la función libera el sistema y el servidor realiza una llamada posterior mientras la variable "curHTTP.callbackPos" sea distinta de cero.

```
//TGF. Rellena una tabla con informacion sobre las GPDAIOs existentes en el sistema
void HTTPPrint_GPDAIOlist(void) {
    WORD len;
    static BYTE cadena[128], *pCadena;
    static UINT16 GPDAIOindex, colIndex, contaGPDAIO;
    BYTE nodeIndex, GPDAIONodeIndex;

    // Inicializacion de datos en la primera llamada
    if(curHTTP.callbackPos == 0u)
    {
        GPDAIOindex = 0;          // Indice de la lista de nodos
        //Busca nodos configurados en la lista. Si llega al
        while(!GPDAIO[GPDAIOindex].node && GPDAIOindex<MAX_SYSTEM_GPDAIO)
            GPDAIOindex++;
        if (GPDAIOindex==MAX_SYSTEM_GPDAIO)
            return;

        curHTTP.callbackPos = 1; // Se pondra a cero cuando hayan salido todos los datos
        cadena[0]=0;
        pCadena = cadena;
        colIndex = 0;           // Indice de columna que estamos representando.
        contaGPDAIO = 0;
    }

    // Determine how many bytes we can write
    len = TCPIsPutReady(sktHTTP);

    while(len && curHTTP.callbackPos)
    {
        if(!(*pCadena))         // Si el puntero no esta apuntando a ningun sitio, se le asigna una
                                // nueva cadena que transmitir
        {
            nodeIndex = 0;
            while(WSN.node[nodeIndex] != GPDAIO[GPDAIOindex].node)
                nodeIndex++;

            switch(colIndex)
            {
                case 0: // Numero de GPDAIO
                    sprintf((char*)cadena, "<tr><td><div align=\"center\">%u</div></td>\n", +
                        contaGPDAIO);
                    colIndex++;
                    break;

                case 1:// Nombre
                    sprintf((char*)cadena, "<td><div>%s</div></td>\n",
                        GPDAIO[GPDAIOindex].name);
                    colIndex++;
                    break;

                case 2:// Valor
                    if( GPDAIO[GPDAIOindex].type == DIGITAL_INPUT ||
```

```

        GPDAIO[GPDAIOindex].type == DIGITAL_OUTPUT
    )
    {
        if (GPDAIO[GPDAIOindex].val == IO_ON)
            strcpy((char*)cadena, "<td><div><div align=\"center\">ON</div></div></td>");
        else
            strcpy((char*)cadena, "<td><div><div align=\"center\">OFF</div></div></td>");
    }
    else if (GPDAIO[GPDAIOindex].type == ANALOG_INPUT)
    {
        sprintf((char*)cadena, "<td><div><div align=\"center\">%3.1f%s</div></div></td>", (double)GPDAIO[GPDAIOindex].val,
            GPDAIO[GPDAIOindex].units);
    }
    else if (GPDAIO[GPDAIOindex].type == PWM_OUTPUT)
    {
        sprintf((char*)cadena, "<td><div><div align=\"center\">%u%%</div></div></td>", (int)GPDAIO[GPDAIOindex].val);
    }
    colIndex++;
    break;

case 3:// Tipo
    if (GPDAIO[GPDAIOindex].type == DIGITAL_INPUT)
        strcpy((char*)cadena, "<td><div><div align=\"center\">Digital Input</div></div></td>");
    else if (GPDAIO[GPDAIOindex].type == DIGITAL_OUTPUT)
        strcpy((char*)cadena, "<td><div><div align=\"center\">Digital Output</div></div></td>");
    else if (GPDAIO[GPDAIOindex].type == PWM_OUTPUT)
        strcpy((char*)cadena, "<td><div><div align=\"center\">PWM Output</div></div></td>");
    else if (GPDAIO[GPDAIOindex].type == ANALOG_INPUT)
        strcpy((char*)cadena, "<td><div><div align=\"center\">Analog Input</div></div></td>");
    colIndex++;
    break;

case 4: // Nombre del nodo y link de configuracion
    if (*(WSN_NODE_TYPE*)GPDAIO[GPDAIOindex].node == ETRX2_NODETYPE)
    {
        sprintf((char*)cadena, "<td><div align=\"center\"><a href=
            \"nodeconfig.htm?nodeID=%u\">%s</a></div></td>\n", nodeIndex,
            ((ETRX2_NODE*)GPDAIO[GPDAIOindex].node)->name);
    }
    colIndex++;
    break;

case 5: // Link configuracion del GPDAIO
    if (*(WSN_NODE_TYPE*)GPDAIO[GPDAIOindex].node == ETRX2_NODETYPE)
    {
        GPDAIONodeIndex = (GPDAIO_OBJECT*)GPDAIO[GPDAIOindex].nodeGPDAIO -
            ((ETRX2_NODE*)WSN.node[nodeIndex])->GPDAIO;

        sprintf((char*)cadena, "<td><div align=\"center\"><a href=
            \"GPDAIOconfig.htm?nodeID=%u&IO=%u&HWtype=%u&name=%s&type=
            %u&dInPol=%u&dOutPol=%u&unit=%s&umin=%f&umax=%f&smin=%f&smax=
            %f&noSave=\">Edit</a></div></td>",
            nodeIndex,
            GPDAIONodeIndex,
            (unsigned int)ETRX2_GPDAIO_TYPE_DEFINITION[GPDAIONodeIndex],
            GPDAIO[GPDAIOindex].name,
            (unsigned int)GPDAIO[GPDAIOindex].type,
            (unsigned int)GPDAIO[GPDAIOindex].digitalInPolarityInverse,
            (unsigned int)GPDAIO[GPDAIOindex].digitalOutPolarityInverse,
            GPDAIO[GPDAIOindex].units,
            (double)GPDAIO[GPDAIOindex].unitMin,
            (double)GPDAIO[GPDAIOindex].unitMax,
            (double)GPDAIO[GPDAIOindex].signalMin,
            (double)GPDAIO[GPDAIOindex].signalMax
        );
    }
    colIndex=0; //Indice de columnas a cero
    GPDAIOindex++;

```

```

        while(!GPDAIO[GPDAIOindex].node && GPDAIOindex<MAX_SYSTEM_GPDAIO)
            GPDAIOindex++;
        break;
    }
    pCadena = cadena;
}

while(len && *pCadena) // Mientras existan datos que enviar y hueco para ponerlos,
                        // pues se envian
    len -= TCPPut(sktHTTP, (BYTE)*pCadena++);

// Si ha terminado de transmitir la cadena y el indice ha llegado al final, esta
// funcion no se llama mas.
if(!(*pCadena) && GPDAIOindex==MAX_SYSTEM_GPDAIO)
    curHTTP.callbackPos=0;
}

return;
}

```

IV.4. Página Web con funciones JavaScript y variables dinámicas

A continuación se muestra un ejemplo de código HTML con llamadas a variables dinámicas tipo ~variable~ y con ejecución de funciones JavaScript. También existen fragmentos de código repetidos en la Web (ej. desplegables con valores de *TimeOut*), agrupado en archivos tipo .inc (include), que son llamados siguiendo el mismo formato que variables dinámicas (~inc:proStTimeOut.inc~).

Código correspondiente a la página "proconfig.htm" (Configuración de Procesos, ver sección "II. 5.1.c. Process Config").

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<title>FJ ESPADA - TFG</title>
<link href="/mchp.css" rel="stylesheet" type="text/css" />
<script src="/mchp.js" type="text/javascript"></script>
<script src="/jsEspada.js" type="text/javascript"></script>
</head>

<body>
<div id="shadow-one">
    <div id="shadow-two">
        <div id="shadow-three">
            <div id="shadow-four">
                <div id="page">
                    <div style="padding:0 0 5px 5px"></div>
                    <div id="title">
                        <div class="right">~config_hostname~</div>
                        <span id="hello">&nbsp;</span></div>
                    <div id="menu"> <a href="/index.htm">System Overview</a> <a href="/
                        prolist.htm">Process</a> <a href="/GPDAIOlist.htm">GPDAIO</a> <a href="/
                        nodelist.htm">Wireless Nodes</a> <a href="/vtconfig.htm">Visualization
                        Terminals</a> <a href="/protect/config.htm">Network Configuration</a> <a
                        href="/email">E-mail</a> <a href="/dyndns">Dynamic DNS</a> </div>
                    <div id="content">
                        <h1>Process Config</h1>
                        <form action="proconfig.htm" method="post" id="prodata">
                            <p>
                                <input type="hidden" name="proID" id="iproID"/>
                                <input type="hidden" name="noSave" id="inoSave" value="1"/>
                                <label for="iname">Process name:</label>
                                <input type="text" name="name" id="iname" />
                            </p>
                            <p>
                                <label for="select">Status:</label>
                                <select name="stasel" id="istasel" onchange="checkSel();">
                                    <option value="0" selected="selected">Ready</option>
                                    <option value="1">Start</option>
                                    <option value="2">Stop</option>
                                    <option value="3">Blocked</option>
                                </select>
                            </p>
                        </form>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>

```

```

</select>
</p>
<div id="reaBox" class="examplebox">
  <h2>READY STATUS</h2>
  <p>
    <input type="checkbox" name="reaemail" id="ireaemail" />
    Send e-mail notification when input condition </p>
  <table width="95%" border="0" align="center" cellpadding="0" id="status0">
    <tr>
      <td colspan="3" align="left" bgcolor="#2099FE"><strong>INPUT CONDITION</strong></td>
    </tr>
    ~readyInputTableContent~
  </table>
  <p>&nbsp;</p>
  <table width="95%" border="0" align="center" cellpadding="0" id="out0">
    <tr>
      <td bgcolor="#2099FE"><strong>OUTPUT STATUS</strong></td>
    </tr>
    ~readyOutputTableContent~
  </table>
</div>
<div id="staBox" class="examplebox">
  <h2>START STATUS</h2>
  <p>
    <input type="checkbox" name="staemail" id="istaemail" />
    Send e-mail notification when input condition </p>
  <p>
    <label for="timeStart">Status Time Out:</label>
    <select name="timeStart" id="itimeStart">

      ~inc:proStTimeOut.inc~

    </select>
  </p>
  <p></p>
  <table width="95%" border="0" align="center" cellpadding="0" id="status1">
    <tr>
      <td colspan="3" bgcolor="#2099FE"><strong>INPUT CONDITION</strong></td>
    </tr>
    ~startInputTableContent~
  </table>
  <p>&nbsp;</p>
  <table width="95%" border="0" cellpadding="0" align="center" id="out1">
    <tr>
      <td bgcolor="#2098FF"><strong>OUTPUT STATUS</strong></td>
    </tr>
    ~startOutputTableContent~
  </table>
</div>
<div id="stoBox" class="examplebox">
  <h2>STOP STATUS</h2>
  <p>
    <input type="checkbox" name="stoemail" id="istoemail" />
    Send e-mail notification when input condition </p>
  <p>
    <label for="timeStop">Status Time Out:</label>
    <select name="timeStop" id="itimeStop">

      ~inc:proStTimeOut.inc~

    </select>
  </p>
  <p></p>
  <table width="95%" border="0" cellpadding="0" align="center" id="status2">
    <tr>
      <td colspan="3" bgcolor="#2099FE"><strong>INPUT CONDITION</strong></td>
    </tr>
    ~stopInputTableContent~
  </table>
  <p>&nbsp;</p>
  <table width="95%" border="0" cellpadding="0" align="center" id="out2">
    <tr>
      <td bgcolor="#2098FF"><strong>OUTPUT STATUS</strong></td>
    </tr>

```

```

        ~stopOutputTableContent~
    </table>
</div>
<div id="bloBox" class="examplebox">
    <h2>BLOCKED STATUS</h2>
    <p>
        <input type="checkbox" name="bloemail" id="ibloemail" />
        Send e-mail notification when input condition </p>
    <table width="95%" border="0" cellpadding="0" align="center" id="status3">
        <tr>
            <td colspan="3" bgcolor="#2099FE"><strong>INPUT CONDITION</strong></td>
        </tr>
        ~blockedInputTableContent~
    </table>
    <p>&nbsp;</p>
    <table width="95%" border="0" cellpadding="0" align="center" id="out3">
        <tr>
            <td bgcolor="#2098FF"><strong>OUTPUT STATUS</strong></td>
        </tr>
        ~blockedOutputTableContent~
    </table>
</div>
<p align="center">
    <table width="90%" border="0" align="center">
        <tr>
            <td><div align="center">
                <input type="button" value="Cancel" onclick="window.location.href='/'>
                prolist.htm"/>
            </div></td>
            <td><div align="center">
                <input type="button" value="Save" onclick="saveAndSubmit()"/>
            </div></td>
        </tr>
    </table>
</p>
</form>
<p> </p>
</div>
<script type="text/javascript">
<!--
var cellColor = ["#E8F2FF", "Aqua", "Aquamarine", "Beige"];

function increaseLevel(table,index)
{
    var level = Number(document.getElementById('i'+table+'leve'+index).value);
    if(level<3)
        level = level+1;
    document.getElementById('i'+table+'leve'+index).value = level.toString();

    document.getElementById('i'+table+'tabu'+index).innerHTML = "";
    for(j=0; j<level;j++)
        document.getElementById('i'+table+'tabu'+index).innerHTML += "&nbsp;";

    var x = document.getElementById('status'+table).getElementsByName("td");
    x[3*index+2].style.backgroundColor = cellColor[level];

    if(index>0)
        x[3*(index-1)+3].style.backgroundColor = cellColor[level];
}

function decreaseLevel(table,index)
{
    var level = Number(document.getElementById('i'+table+'leve'+index).value);
    if(level>0)
        level = level-1;
    document.getElementById('i'+table+'leve'+index).value = level.toString();

    document.getElementById('i'+table+'tabu'+index).innerHTML = "";
    for(j=0; j<level;j++)
        document.getElementById('i'+table+'tabu'+index).innerHTML += "&nbsp;";

    var x = document.getElementById('status'+table).getElementsByName("td");
    x[3*index+2].style.backgroundColor = cellColor[level];

    if(index>0)
        x[3*(index-1)+3].style.backgroundColor = cellColor[level];
}

```

```

function checkSel()
{
    var sel = document.getElementById('istasel').value;

    document.getElementById('reaBox').style.display = 'none';
    document.getElementById('staBox').style.display = 'none';
    document.getElementById('stoBox').style.display = 'none';
    document.getElementById('bloBox').style.display = 'none';

    switch(Number(sel))
    {
        case 0:
            document.getElementById('reaBox').style.display = 'block';
            break;

        case 1:
            document.getElementById('staBox').style.display = 'block';
            break;

        case 2:
            document.getElementById('stoBox').style.display = 'block';
            break;

        case 3:
            document.getElementById('bloBox').style.display = 'block';
            break;
    }
}

function saveAndSubmit()
{
    document.getElementById('inoSave').value = 0;
    document.getElementById('prodata').submit();
}

readURLfields();
getField('proID');
getField('name');

var status = params.statusN;
var inNumber = params.inN;
var outNumber = params.outN;

function setInitialValues()
{
    document.getElementById('ireaemail').checked = ~reaemail~;
    document.getElementById('istaemail').checked = ~staemail~;
    document.getElementById('istoemail').checked = ~stoemail~;
    document.getElementById('ibloemail').checked = ~bloemail~;

    document.getElementById('itimeStart').value = params.timeStart;
    document.getElementById('itimeStop').value = params.timeStop;
    document.getElementById('istasel').value = params.stasel;

    checkSel();

    for(var sta=0; sta<status; sta++)
    {
        for(var i=0; i<inNumber; i++)
        {
            var y =
            document.getElementById('status'+sta).getElementsByName("td");
            y[3*i+2].style.backgroundColor =
            cellColor[Number(document.getElementById('i'+sta+'leve'+i).value)];

            if(i>0)
                y[3*(i-1)+3].style.backgroundColor =
                cellColor[Number(document.getElementById('i'+sta
                +'leve'+i).value)];

            document.getElementById('i'+sta+'tabu'+i).innerHTML= "";
            for(t=0; t<Number(document.getElementById('i'+sta+'leve'+i).value);t++)
                document.getElementById('i'+sta+'tabu'+i).innerHTML +=
                "&emsp;";
        }
    }
}

```



```
    }  
}  
  
setInitialValues();  
  
/-->  
</script>  
    <div class="spacer">&nbsp;</div>  
    <div id="footer">Trabajo Final de Grado - Francisco Jos&eacute; Espada Moreno -  
        Sept. 2014</div>  
    </div>  
</div>  
</div>  
</div>  
</div>  
</body>  
</html>
```


V. PRESUPUESTO

V.1. Diseño y Desarrollo Hardware y Software

Herramientas Desarrollo						
Ref	Concepto		Coste Unitario	Un	Importe	% cargo proyecto
1.1. 1	Licencia Altium 2010		4.000,00	1	4.000,00	5%
1.1. 2	Licencia AutoCad		2.500,00	1	2.500,00	5%
1.1. 3	PC sobremesa + pantalla 27"		1.100,00	1	1.100,00	5%
1.1. 4	Osciloscopio Tektronix TBS1052B		420,00	1	420,00	5%
1.1. 5	Polímetro Fluke 187v		599,00	1	599,00	2%
1.1. 6	Herramientas y pequeño material		300,00	1	300,00	100%
1.1. 7	Programador ICD3		150,00	2	300,00	20%
1.1. 8	Convertor TTL-USB		30,00	2	60,00	50%
1.1. 9	Tarjeta de Desarrollo PIC32 Ethernet Starter Kit		70,00	1	70,00	100%
1.1. 10	Tarjeta de Expansión IO Expansion Board		70,00	1	70,00	100%
1.1. 11	Tarjeta Pigtail SD card		30,00	1	30,00	100%
1.1. 12	AP WiFi básico		30,00	1	30,00	100%
1.1. 13	Tableta 7" WVGA WiFi		59,00	2	118,00	100%
1.1. 14	Ordenador MacBook Pro 15" Retina		1.700,00	1	1.700,00	10%
1.1. 14	Ratón + Teclado inalámbricos		140,00	1	140,00	10%
1.1	SUBTOTAL					1.304,98 €

Horas Diseño y Desarrollo				
Ref	Concepto		Coste Unitario	Un
1.2. 1	Ingeniero Desarrollo		60,00	320
1.2	SUBTOTAL			19.200,00 €

Otros Conceptos Diseño y Desarrollo					
Ref		Concepto	Coste Unitario	Un	Importe
1.3.	1	Material de oficina	250,00	1	250,00
1.3.	2	Alquiler de oficinas y despachos (mes)	600,00	2	1.200,00
1.3.	3	Otros gastos desarrollo actividad	500,00	1	500,00
1.3		SUBTOTAL			1.950,00 €

COSTE TOTAL DISEÑO Y DESARROLLO		
Ref	Concepto	Importe
1.	1 Herramientas Desarrollo	1.304,98
1.	2 Horas Diseño y Desarrollo	19.200,00
1.	3 Otros Conceptos Diseño y Desarrollo	1.950,00
	Gastos Generales (30%, sobre 1.1, 1.2 y 1.3)	6.736,49
	Beneficio (50% sobre 1.2)	9.600,00
1	TOTAL COSTE DISEÑO Y DESARROLLO	38.791,47 €

V.2. Fabricación de Prototipos

Fabricación Unidad Coordinadora (precio unidad)				
Ref		Concepto	Coste Unitario	Un Importe
2.1.	1	Fabricación de circuitos impresos	5,00	1 5,00
2.1.	2	Coste BOM	80,00	1 80,00
2.1.	3	Ensamblado de componentes en PCB	7,00	1 7,00
2.1	SUBTOTAL			92,00 €

Fabricación Nodo Inalámbrico (precio unidad)				
Ref		Concepto	Coste Unitario	Un Importe
2.2.	1	Fabricación de circuitos impresos	3,00	1 3,00
2.2.	2	Coste BOM	20,00	1 20,00
2.2.	3	Ensamblado de componentes en PCB	1,50	1 1,50
2.2	SUBTOTAL			24,50 €

Precio Final Pack Prototipo (1 unidad coordinadora + 20 nodos inalámbricos)		
Ref	Concepto	Coste Unitario
2.3	PACK UNIDAD COORDINADORA + 20 NODOS INALÁMBRICOS	582,00 €

V.3. Coste Total: Desarrollo + Prototipos

Coste Total Desarrollo + Prototipos		
Ref	Concepto	Beneficio unitario
3.1. 1	TOTAL COSTE DISEÑO Y DESARROLLO	38.791,47 €
3.1. 2	PACK UNIDAD COORDINADORA + 20 NODOS INALÁMBRICOS	582,00 €
	SUBTOTAL AI	39.373,47 €
3.1. 3	IVA (21%)	8.268,43 €
3.1	COSTE TOTAL DESARROLLO + PROTOTIPOS	47.641,90 €

VI. LINKS Y BIBLIOGRAFÍA

- (1) A. Soomro, D. Cavalcanti, "Opportunities & Challenges using WPAN and WLAN Technologies in Medical Environments", IEEE Communications Magazine, vol. 45, no.2, Feb 2007, page 114-122.
- (2) Cengiz Gezer, and Chiara Buratti "A ZigBee Smart Energy Implementation for Energy Efficient Buildings" DEIS, University of Bologna, Italy.
- (3) eDIANA, Artemis Project, <http://www.artemis-ediana.eu/>
- (4) Telegesis, "ZigBee and Internet of Things" "http://www.telegesis.com/our_markets/internet_of_things.htm
- (5) Evolution of the ZigBee Market. iTers News. 2013.
- (6) HomeRF. en.m.wikipedia.org/wiki/HomeRF
- (7) MiWi. www.microchip.com/miwi/
- (8) RAINOW (RFCR-CT-2005-00003). Informe Final del Proyecto.
- (9) IEEE 802.15. <http://standards.ieee.org/about/get/802/802.15.html>
- (10) ZigBee Tutorial. <http://www.ifn.et.tu-dresden.de>
- (11) Telegesis. Application Note: TG-APP-ETRX2Power-100. v.1.0
- (12) ZigBee alliance. www.zigbee.org
- (13) Telegesis. www.telegesis.com
- (14) DynDNS. www.dyndns.com
- (15) No-IP. www.no-ip.com
- (16) DNS-O-Matic. www.dnsomatic.com
- (17) Silicon Labs. www.silabs.com
- (18) Librerías jQuery. jquery.com